ETSI TS 102 542 V1.2.1 (2008-04)

Technical Specification

Digital Video Broadcasting (DVB); Guidelines for the implementation of DVB-IP Phase 1 specifications

European Broadcasting Union

Union Européenne de Radio-Télévision

EBU·UER

DVB

Digital Video Broadcasting

ETSI

Reference

RTS/JTC-DVB-219

Keywords

broadcasting, digital, DVB, IP, TV, video

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

Please note that the present document is a revision to TR 102 542, and has been converted to a TS because the language used in the document is akin to that of a TS.

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

Founded in September 1993, the DVB Project is a market-led consortium of public and private sector organizations in the television industry. Its aim is to establish the framework for the introduction of MPEG-2 based digital television services. Now comprising over 200 organizations from more than 25 countries around the world, DVB fosters market-led systems, which meet the real needs, and economic circumstances, of the consumer electronics and the broadcast industry.

1 Scope

The present document is designed as a companion document to help implement the DVB-IP Phase 1 version 3: Transport of MPEG2-TS Based DVB Services over IP Based Networks [1], which is referred to as the Handbook. The present document is organized in separate sections in the order of the boot-up sequence of the HNED rather than in the same section structure as the Handbook. Each clause deals with a specific aspect of the DVB-IP technology, and offers explanations and examples not found in the Handbook. Additionally, it provides guidelines to implement the Broadband Content Guide (BCG) specification [3].

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] ETSI TS 102 034 (V1.3.1): "Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks".

[2] ETSI TS 101 154 (V1.8.1): "Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream".

[3] ETSI TS 102 539 (V1.2.1): "Digital Video Broadcasting (DVB); Carriage of Broadband Content Guide (BCG) information over Internet Protocol (IP)".

[4] ETSI TS 102 822-2 (V1.3.1): "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 2: System description".

[5] ETSI TS 102 822-6-1 (V1.3.1): "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime"); Part 6: Delivery of metadata over a bi-directional network; Sub-part 1: Service and transport".

[6] ETSI TS 126 346: "Universal Mobile Telecommunications System (UMTS); Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs (3GPP TS 26.346 Release 7)".

[7] SMPTE Specification 2022-1: "Forward Error Correction for Real-time Video/Audio Transport Over IP Networks".

[8] DVB BlueBooks A109: "DVB-HN (Home Network) Reference Model Phase 1".

[9] ETSI TS 102 323: "Digital Video Broadcasting (DVB); Carriage and signalling of TV-Anytime information in DVB transport streams".

[10] ETSI TS 102 005: "Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in DVB services delivered directly over IP protocols".

2.2 Informative references

[11] IETF RFC 3927: "Dynamic Configuration of IPv4 Link-Local Addresses".

[12] IETF RFC 3203: "DHCP reconfigure extension".

[13] IEEE P802.11-REVma/D6.0, 2006: Unapproved Draft Standard for Information Technology-Telecommunications and information exchange between systems- Local and metropolitan area network- Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

NOTE: This document reflects the combining of the 2003 Edition of 802.11 plus the 802.11g, 802.11h, 802.11i and 802.11j Amendments) (Revision of IEEE Std 802.11-1999).

[14] IEEE 802.1d (2004) "IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) Bridges".

[15] IETF RFC 3376: "Internet Group Management Protocol, Version 3".

[16] IETF RFC 1112: "Host extensions for IP multicasting".

3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ALG	Application Level Gateway
AVC	Advanced Video Coding
BCG	Broadband Content Guide
BiM	Binary MPEG Format for XML
CRLF	Carriage Return Line Feed
DHCP	Dynamic Host Configuration Protocol
DNG	Digital Network Gateway
DSCP	Differentiated Services CodePoint
DSL	Digital Subscriber Line
DTD	Document Type Declaration
DVB	Digital Video Broadcasting
DVBSTP	DVB SD&S Transport Protocol
HNED	Home Network End Device
HTTP	Hyper Text Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPI	IP Infrastructure
LAN	Local Area Network
LCN	Logical Channel Numbers
MPEG	Moving Picture Experts Group
MPTS	Multi Program Transport Stream
NAK/NACK	Negative ACKnowledge
RAM	Random Access Memory

RFC	Request For Comments
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SD&S	Service Discovery and Selection
SI	Service Information
SOAP	Simple Object Access Protocol
SPTS	Single Program Transport Stream
SSL	Secure Socket Layer
TS	Transport Stream
UDP	User Datagram Protocol
XML	eXtensible Markup Language

4 Background to the Scenarios

The following figure shows the Home Reference Model for the DVB-IP phase 1, taken from the Handbook (see [1], clause 4.1.2).

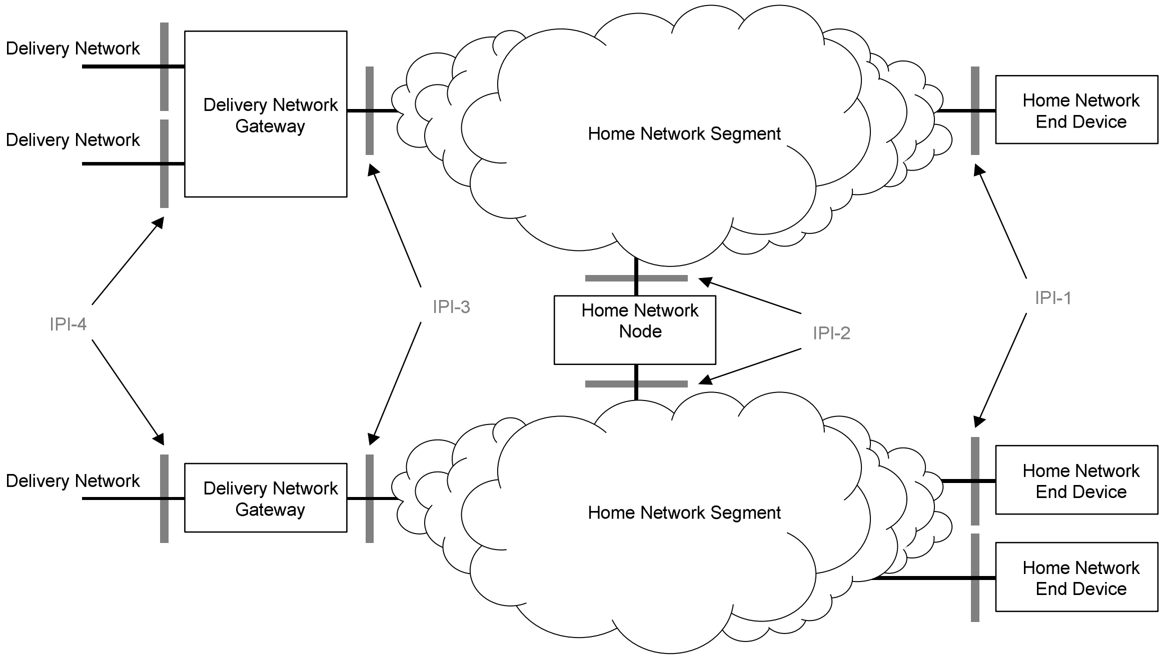

Figure 1: Home Reference Model (from TS 102 034 [1])

Figure 1 and the current version of the DVB-IP Handbook [1] focuses only on the delivery of DVB-IP services over broadband delivery networks. DVB is working on enhanced home networking functionality which will for example allow an end user to access DVB content from several devices in the home. The Home Network Reference Model for this approach is provided in [8]. The protocols and functions to support this Home Network Reference Model will be defined in upcoming specifications and therefore not covered in the current version of the present document

The Handbook only specifies the IPI-1 interface at the HNED (Home Network End Device). However, the specification of the IPI-1 interface also defines characteristics of the Home Network Segment between the HNED and the DNG, and in some cases what the DNG must deliver.

The Handbook intentionally does not attempt to specify where particular servers need to reside, for example the DHCP server. This means that no protocol is defined to operate solely on the home network segment. It also means that operation of one HNED is completely independent of the operation of another HNED in the same Home Network. Although multiple HNEDs in the same Home Network will share IP connectivity, there is no specific protocol defined in the handbook to allow them to exchange messages, or even know about the presence of each other.

The DVB-IP Handbook does not currently define the interface IPI-2 so any routing or translation scenario that may be required for interworking between Home Network Segments is outside of the scope of Phase 1 of the handbook. This means that many HNEDs can be connected to a single DNG, but multiple DNGs connected on the same network segment is not allowed.

5 Turning on and Booting an HNED

The best way to describe how the DVB-IP Handbook can be used is to go through what happens when you turn on an HNED. There are a number of steps in order to have:

- Physical/MAC Layer Connection.
- IP Layer connectivity via obtaining an IP Address.
- Network Provisioning (optional).
- Connection to the SD&S servers.
- Discovery of BCG information (optional).
- Content Selection.
- Streaming of the video content.

Network Provisioning is optional and is dealt with in a separate clause.

5.1 Physical/MAC Layer Connection

The physical and the link layers need to come up before anything else happens. The DVB-IP handbook requires a IEEE 802 based MAC layer with priority marking according to IEEE 802.1d [14] within the home network segment. These can be used by the network to help obtain the Quality of Service required for the streamed video content.

5.2 IP Layer connectivity via obtaining an IP Address

Once the link layer comes up, the HNED obtains the IP address from a DHCP server with the DVB mandatory DHCP options. The handbook specifies the minimum DHCP options required to allow the DHCP server to be simple enough to fit into a DNG or other product on the home network segment.

DHCP does not currently specify a way to co-ordinate the address pools of multiple DHCP servers on a network. The DHCP client simply takes the first address offered to it but, normally, the closest available server. This means that multiple DHCP servers cannot be used on the same network to serve the HNED.

The IP address assigned by the DHCP server will be different for each HNED on the same home network segment, but will be part of the same IP subnet. The use of private or public IP address space and size of the subnet mask is at the discretion of the Network Service Provider.

NOTE: **zero-configuration mechanism:**
Whilst the DVB-IP specification proposes two ways for HNEDs to get an IP address: DHCP server or via RFC 3927 [11] (IETF zero configuration mechanism), DHCP server is the normal way. It is expected that the RFC 3927 [11] is only to be used in emergency where the DHCP server is down for some short-term reason. Running in zero-conf mode provides none or very little connectivity. Basically, the HNED does not have knowledge of a gateway device to send messages to external servers, so the only possible scenario is to connect to multicast streams (provided the DNG let IGMP messages flow over to the outside): first an SD&S stream then a live TV stream.

5.2.1 Location of the DHCP Server

The DHCP server can be located in the home or in the access network. If it is in the home, it will likely be on the DNG, a scenario typical of DSL. The most popular means of address assignment is to have the home in a private IP address space whilst the public interface has an IP address given by the network operator as shown in figure 2. The DNG uses Network Address Translation to change the IP addresses of the data from public to/from private address spaces.

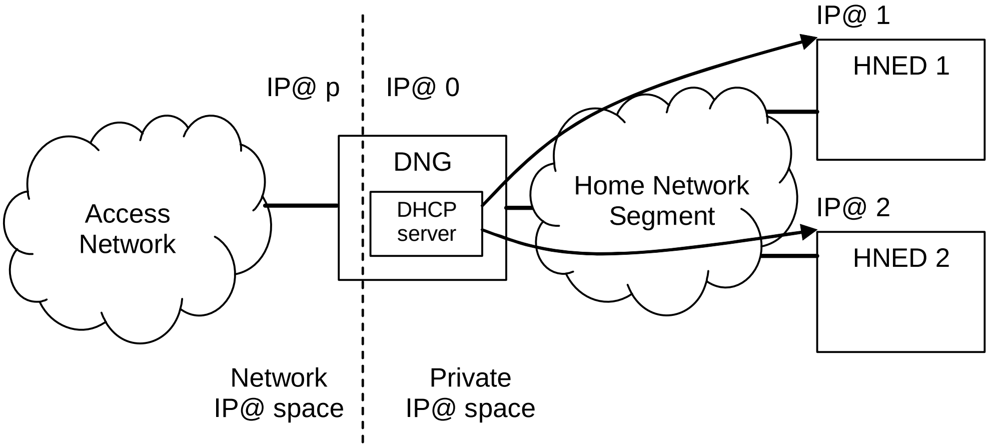

Figure 2: Home Network with local DHCP server

The DHCP server can be located on the external network, typical of some DSL, or most cable or Ethernet to the Home deployments. The DNG then acts as a bridge or DHCP "relay" to relay the DHCP messages to the external DHCP server as shown in figure 3. Please be aware that the DVB Class options must be preserved in this case.

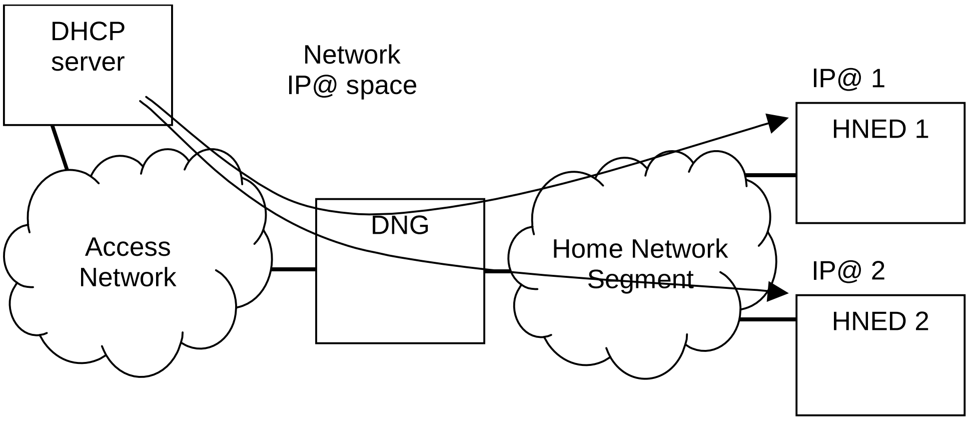

Figure 3: Home Network with remote DHCP server

In order to overcome problems with local DHCP servers and Address Translation, IPTV deployments in DSL networks often connect the HNED to a bridge port of the DNG which directly connects the HNED to the Access Network at the link layer below IP. The HNED is in this case within the IP address space of the Access Network and uses the DHCP server of the Access Network as shown in figure 4. A disadvantage is that the HNED is separate from the Home Network of the user which is connected via routed ports of the DNG.

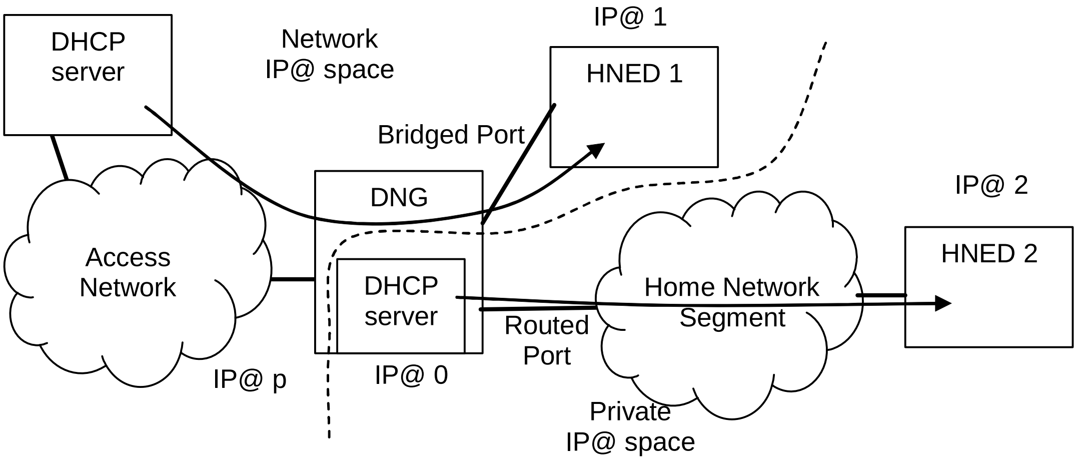

Figure 4: Home Network with remote DHCP server

5.2.2 Adding a New DHCP Class Option

The DHCP Class IDs defined in the Handbook are the minimum set needed to support the types of HNEDs originally supported in the commercial and technical requirements. The Handbook allows these attributes to be added to by any DVB member.

The Class ID is meant to help the DHCP server gives the appropriate IP address for the type of HNED. It is an insecure method but, for example, will allow a DHCP server to give a private address to one type of HNED and a public one to another. It should not be manufacturer specific.

Following is the procedure to add a new attribute:

1) Contact the DVB Project Office via the web site or email with the following information:
   - Name of the Class ID.
   - Company name.
   - Contact name, email address and phone number of the legal representative who is the signatory to the request.
   - Contact name, email address and phone number of the technical representative for the request.
   - Technical and Commercial motivation for the request.
2) The DVB Project Office will optionally contact the company.
3) The DVB Project Office will then notify the technical and legal representative of their decision.
4) If the decision is positive then the class ID will be published on the DVB web site and, if possible, in the next maintenance revision of the Handbook.

5.3 Content Discovery

Now that the HNEDs have their IP address, they start looking for the SD&S servers(s) to retrieve the service lists. Figure 5 shows several ordered steps that a HNED walks through to connect to the service providers.

1. DHCP server sets the *siaddr* field for network provisioning
   => SD&S Entry Point Provisioned (see Network Provisioning)

2. DHCP server has *siaddr*=0 but sets the *Domain Name* option
   => DNS SRV to the servers in the Domain Name Option

3. DHCP server has *siaddr*=0 and gives no *Domain Name*
   => DNS SRV to the default server « services.dvb.org »

4. No server can be contacted
   => HNED connects to the multicast address 224.0.23.14

5. Nothing has worked
   => user configures manually the SD&S address

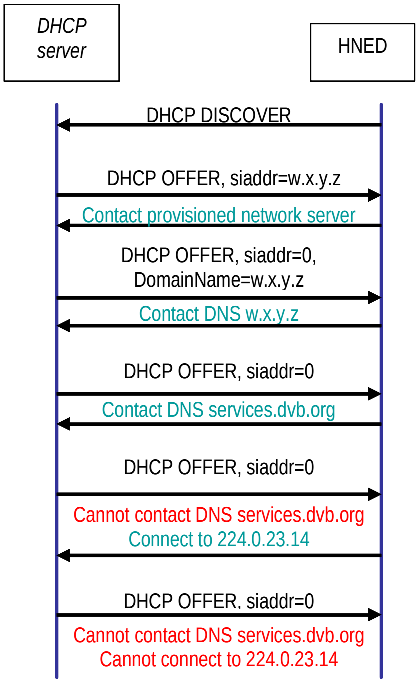
DHCP server
HNED
DHCP DISCOVER
DHCP OFFER, siaddr=w.x.y.z
Contact provisioned network server
DHCP OFFER, siaddr=0,
DomainName=w.x.y.z
Contact DNS w.x.y.z
DHCP OFFER, siaddr=0
Contact DNS services.dvb.org
DHCP OFFER, siaddr=0
Cannot contact DNS services.dvb.org
Connect to 224.0.23.14
DHCP OFFER, siaddr=0
Cannot contact DNS services.dvb.org
Cannot connect to 224.0.23.14

Figure 5: SD&S server Entry Point discovery order

5.3.1 Content Discovery with Local DHCP Server

The number of mechanisms reflects the different topologies of the service provider and in-home networks, and DNGs. For example, current DSL providers use DNGs with DHCP servers that sometimes do not support network provisioning or the DHCP Domain Name Option, so it is possible that the DHCP server in the DNG in the home will not support steps 1 and 2.

However, the *giaddr* field will be set (it indicates the IP address of the gateway device). This means that with basic NAT feature on the gateway device, step 3 can be performed. The HNED can connect to the default DVB server (HNED 1 in figure 6), or better directly to a specific provider (HNED 2 - this happens when the HNED is coming from the content provider, so it knows the address of its server).

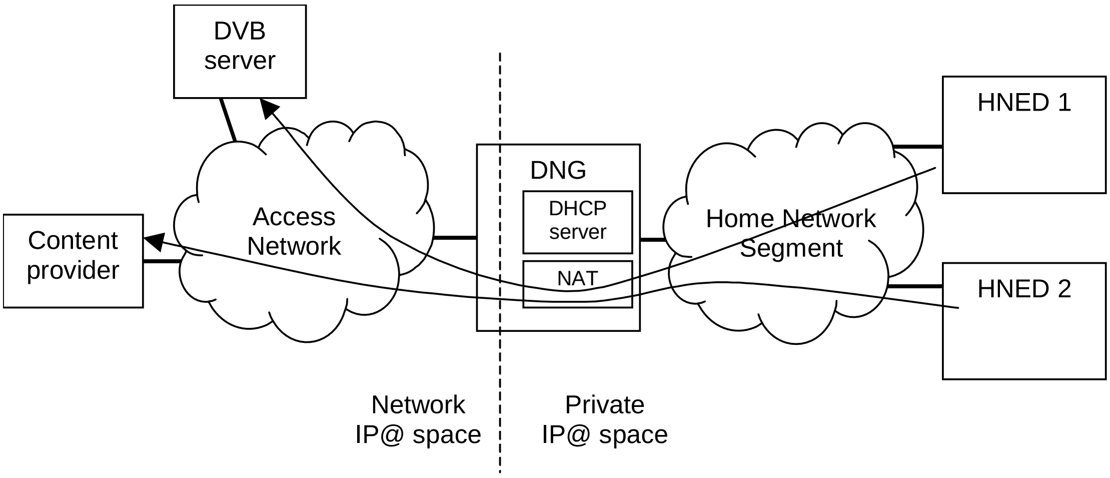

Figure 6: Content discovery with DHCP server

5.3.2 Content Discovery without DHCP Server

Whilst an HNED without a corresponding DHCP server is an abnormal situation, HNEDs may still retrieve the service lists. This is by using the DVB assigned multicast address (step 4 in figure 5). If the DNG forwards the IGMP messages from the HNEDs (this is a broadcast message on the Home Network Segment, so the gateway will receive it), and provided that the DNG forwards incoming multicast packets from the access network into the home, then the DVBSTP stream can be received by the HNED. It will then build the service list based on the content of this stream.

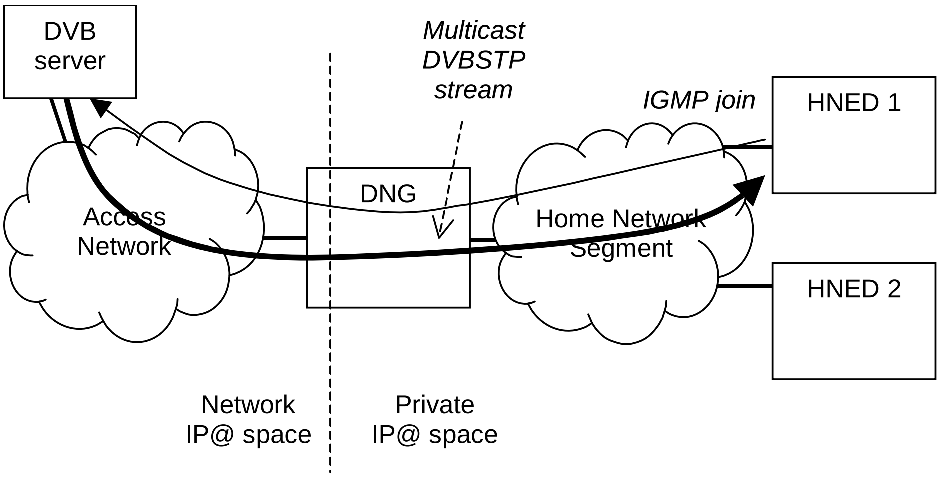

Figure 7: Content discovery without DHCP server

Note that this solution may work for Live Media Broadcast Content only. Live Media Broadcast Content may require only an IGMP message to get the AV multicast stream, without RTSP protocol. Content on Demand will not be possible because it requires RTSP, and the HNED does not know where to send the RTSP message (no gateway identified).

5.4 Content Selection

With DVB-IP phase 1, there are basically three ways to access content:

- Multicast stream selection only.
- Multicast stream selection plus RTSP.
- Unicast stream with RTSP.

The first two steps are for live TV content while the latter is for content on demand or Media Broadcast with Trick Modes services. For Live TV, the RTSP messages are not mandatory; it is perfectly possible for the HNED to just join the corresponding multicast group.

5.4.1 DHCP Server within the Home

The multicast join message is sent on the HN, and the gateway forwards it to the access network. Thus the Live TV stream can be received by the HNED.

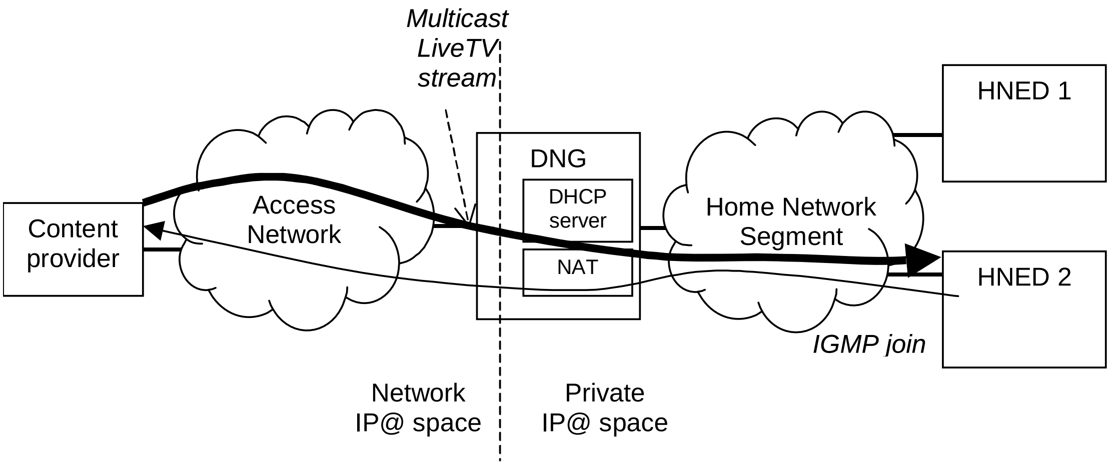

Figure 8:IGMP live content selection with DHCP server

If the RTSP protocol is used, the gateway needs to provide RTSP ALG (Application Level Gateway) feature: this ALG replaces into the RTSP message payload the values of the IP address and UDP port given by the HNED by the public IP address of the gateway and an available UDP port. This RTSP message will be sent before doing the multicast join.

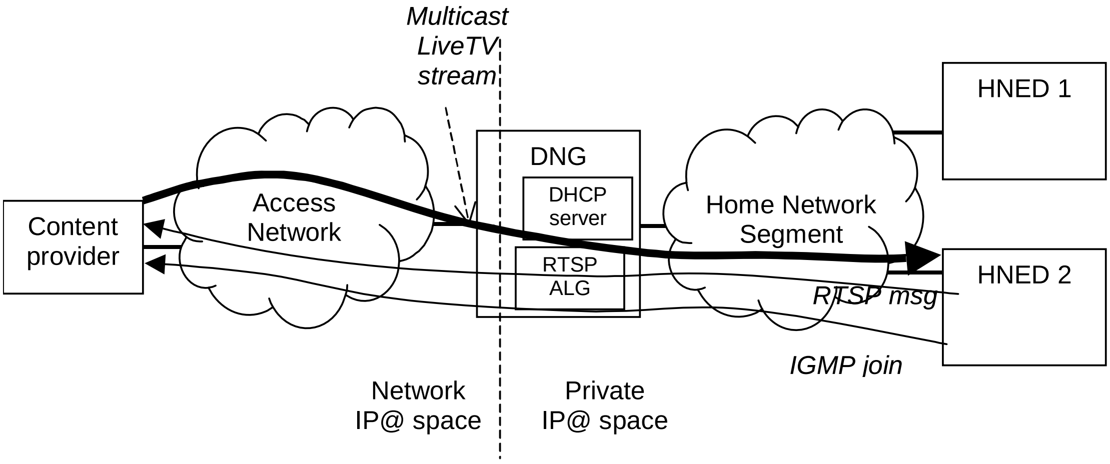

Figure 9: RTSP live content selection with DHCP server

Finally, in case of unicast streaming, no multicast join is necessary but the gateway still needs its RTSP ALG feature.

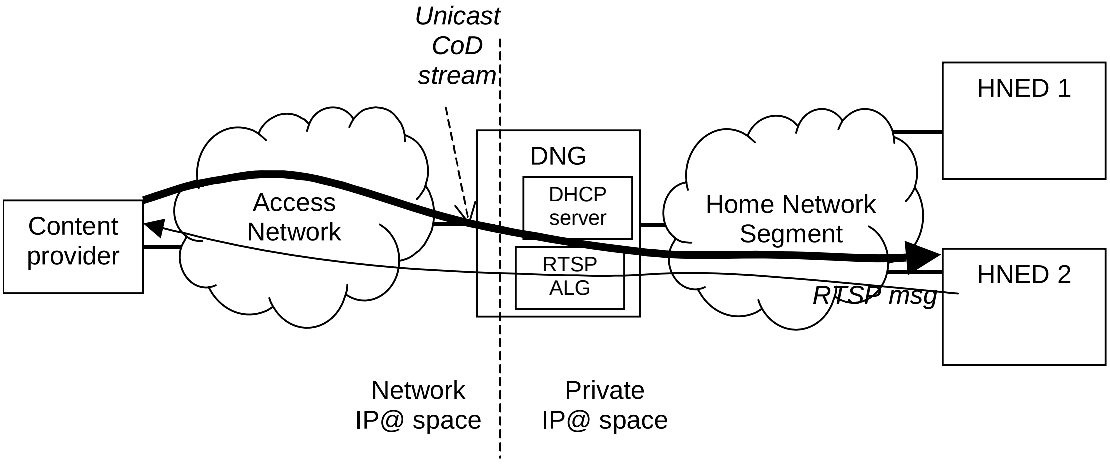

Figure 10: Content on Demand selection

5.4.2 No DHCP Server

Again, as in clause 5.3.2, this may work provided that the DNG forward the multicast report message, and forward incoming multicast packets in the home. The only possibility with this configuration is to connect to a Live TV stream without RTSP protocol.

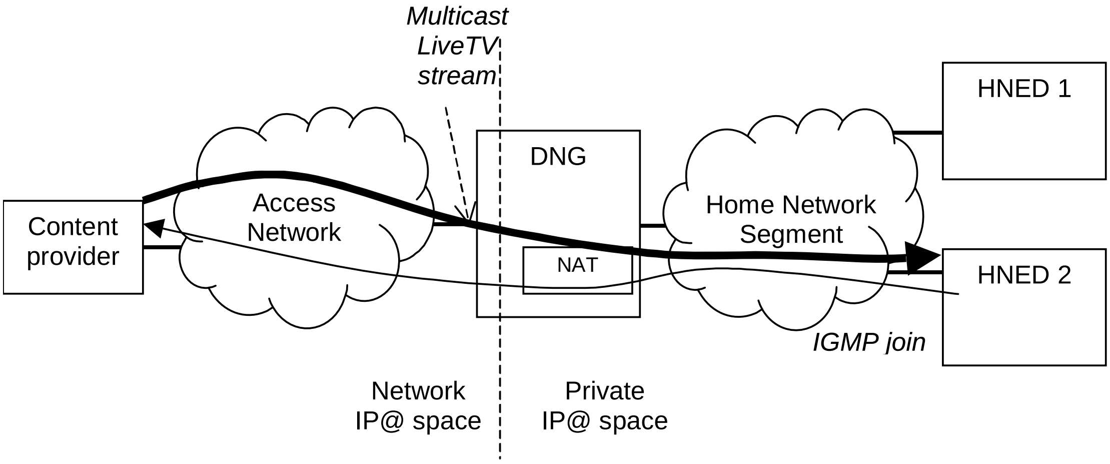

Figure 11: Live content selection without DHCP server

6 SD&S Service Discovery

6.1 Push and Pull modes

Once one of the ways of selecting the Service Discovery entry points has been chosen, the HNED knows the entry points and can access the SD&S server either in multicast or unicast way. For each entry point, the HNED collects the Service Provider Discovery information.

The Service Provider Discovery Information may be (according to the IP address class of the Service Discovery entry point):

- **Multicast (Push model)**: The HNED sends an IGMP Report request to a multicast address in order to subscribe to this multicast group. The content of the "Provider" XML file is carried by the **DVBSTP** protocol with a payload id value set to 0x01 (See table 1: Payload ID values of TS 102 034 [1]).

- Retrieved on **request (Pull model).** In this case, the HNED sends a **HTTP request:**

```
'GET /dvb/sdns/sp_discovery?id=ALL HTTP/1.1' CRLF
'Host: ' <host> CRLF
```

or

```
'GET /dvb/sdns/sp_discovery?id=<DomainName> HTTP/1.1' CRLF
'Host: ' <host> CRLF
```

Both models are supported.

The HNED gets thus the Service Providers' list and their Push or Pull offers. The HNED selects the **Push** [Multicast IP address (IGMP), content of XML file carried by DVBSTP] or **Pull** (in this case, it is done through a HTTP request) offers of its Service Provider.

For information, the Payload ID values of the different SD&S services are shown in table 1 of the TS 102 034 [1].

Service discovery information is represented as XML records (examples are given hereafter). In order to be managed efficiently by the HNED, SD&S records are fragmented into a number of smaller units, called Segments. Segments may be transported uncompressed or compressed using BiM.

BiM compression reduces the size of the SD&S XML records significantly, therefore its use is recommended in constrained environments. When the network provider uses BiM compression, it shall also make available uncompressed Segments (in Push, Pull or both modes).

6.2 Strategies for SD&S Service Discovery

6.2.1 Choosing between push and pull modes

Multicast and HTTP sources for SD&S information are extracted first from the Entry Point Discovery process and then from Service Provider descriptions. For each of these steps, the HNED may find either only one transport model provided or both. When both are provided, they convey exactly the same information (XML records) so there is a choice to use one method or the other. Using the HTTP mode has the drawback of increasing the server load proportionally to the number of HNED on the network, which can be millions. Multicast mode has the drawback of a potential delay of 30 seconds in order to scan a complete carrousel cycle. A general recommendation for "fair behaviour" of the HNED would be then to prefer multicast mode when no specific reason asks for an immediate acquisition of some information (which is however never guaranteed) and reserve the use of HTTP for infrequent situations where the application needs to provide up-to-date information quickly to the end user.

6.2.2 Different scenarios regarding transport of multiple segments

6.2.2.1 Finding the segment lists

When using an **entry point address** in pull mode, the HTTP request always sends back the complete service discovery record without segmentation. This can be the complete set for all service providers (request with "`id=ALL`") or a specific service provider record ("`id=<domain name of the service provider>`").

When receiving SD&S information from multicast address(es), the DVBSTP header has a payload ID field and a segment ID field on each packet. These fields allow to capture the list of segments for each payload type by listening to a complete carrousel cycle time.

Additionally service provider records may list the segments that are made available through each announced source (either push or pull). This list is mandatory for pull sources since there is no other way (see note) in this case to get the list. It may be provided for push mode since this allows the HNED to know if it has acquired everything without necessarily waiting for the maximum cycle time.

NOTE: Actually there is a way: send a request to get each possible segment number and see which succeed or fail. But this is not practically feasible with 65 536 possible segment IDs.

6.2.2.2 Filtering service providers in DVBSTP

The DVB-IP specification allows the case where several Service Providers use independent equipments to serve their own SD&S data. It is then possible that several service providers use the same multicast group to send DVBSTP packets containing the descriptions of their offer (**Service Provider Discovery records**). Then because the service providers are not centrally coordinated, they might well choose segment IDs that are the same.

The DVBSTP has a `ServiceProviderID` field, mandatory in this kind of configuration, that is used to signal the service provider identity (in the form of an IP address unique to the service provider). This allows the HNED to sort the received information and build a correct list of segments assigned by each service provider. See [1], clause 5.4.1.3.3.

6.3 Acquisition of Live Channels Services

The acquisition of Live Channels services is performed through the retrieval of the content of the "Package" and "Broadcast" files.

NOTE: Live Channels can also be retrieved thanks to the BCG; this is presented in the BCG clause of the present document.

If there is a Package service, the HNED can collect (via Push or Pull mode) the "service names" of the channels composing its bouquet.

Then, the HNED can access (via Push or Pull mode) the XML file that contains the BroadcastDiscovery structure.

For the Broadcast Discovery Information Record, there are two modes:

- "TS Full SI" (SD&S + DVB-SI):
  It provides only the necessary SD&S information to find available live media broadcast services which have embedded SI. Information on individual services is afterwards acquired from the transport stream itself through classical use of service information as defined in DVB-SI. As even the service name is not provided in SD&S, the HNED needs to connect to all services and parse all SI to build the service list.

- "TS Optional SI" (only SD&S):
  It provides all the necessary SD&S information to create a list of available services with sufficient information for the user to make a choice and gives the necessary information on how to access the service.

In the following part, we consider the "TS Optional SI" mode.

6.4 Complete SD&S example

This clause presents a workable example of SD&S discovery, and shows all different DVB-IP technologies that can be used (packages, FEC, regionalization, media transport)

6.4.1 Service Provider Discovery Record

As an example, the Service Provider discovery record below contains 2 service providers: "Provider1" and "Provider2". Each provider proposes 2 services: a Package service (Payload ID value=5) and a Broadcast service (Payload ID value=2).

	Provider1	**Provider2**
Services	Package (Payload ID value=5)	Package (Payload ID value=5)
	Broadcast (Payload ID value=2)	Broadcast (Payload ID value=2)

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <ServiceProviderDiscovery>
        <ServiceProvider DomainName="provider1.com" LogoURI="0" Version="0">
            <Name Language="ENG">Provider1</Name>
            <Description Language="ENG">Provider1 ADSL TV Offer</Description>
            <Offering>
                <Push Address="224.1.1.5" Port="1234" Source="192.100.100.70">
                    <PayloadId Id="5">
                        <Segment ID="1" Version="2"/>
                    </PayloadId>
                </Push>
            </Offering>
            <Offering>
                <Push Address="224.1.1.2" Port="1234" Source="192.100.100.70">
                    <PayloadId Id="2">
                        <Segment ID="3" Version="2"/>
                    </PayloadId>
                </Push>
            </Offering>
        </ServiceProvider>
        <ServiceProvider DomainName="provider2.com" LogoURI="0" Version="0">
            <Name Language="ENG">Provider2</Name>
            <Description Language="ENG">Provider2 ADSL TV Offer</Description>
            <Offering>
                <Push Address="224.1.1.6" Port="1234" Source="192.100.100.75">
                    <PayloadId Id="5">
                        <Segment ID="0" Version="0"/>
                    </PayloadId>
                </Push>
            </Offering>
            <Offering>
                <Push Address="224.1.1.3" Port="1234" Source="192.100.100.75">
                    <PayloadId Id="2">
                        <Segment ID="2" Version="3"/>
                    </PayloadId>
                </Push>
            </Offering>
        </ServiceProvider>
    </ServiceProviderDiscovery>
</ServiceDiscovery>
```

6.4.2 Package and Broadcast Discovery with Regionalization

As an example, the " Package " file below corresponds to the "Provider1". For this service provider, 2 bouquets are proposed: "Provider1 Bouquet1" and "Provider1 Bouquet2". The bouquet "Provider1 Bouquet1" contains the channels "Channel 2", "Channel 3" and "Channel 5". The bouquet "Provider1 Bouquet2" contains the channels "Channel 7", "Channel 8" and "Channel 9". The provider uses UDP streaming.

Furthermore, the service provider assigns Logical Channel Numbers (LCN) to services described in the SD&S records, as presented in the following table, and also provides availability information.

	Provider1 Bouquet1		Provider1 Bouquet2	
Channels	Channel2	LCN = 1	Channel 7	LCN = 2
	Channel3	LCN = 3	Channel 8	LCN = 4
	Channel5	LCN = 6	Channel 9	LCN = 5

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <PackageDiscovery DomainName="provider1.com" Version="0">
        <Package Id="1">
            <PackageName Language="ENG">Provider1 Bouquet1</PackageName>
            <Service>
                <TextualID ServiceName="Channel2"/>
                <LogicalChannelNumber="1"/>
            </Service>
            <Service>
                <TextualID ServiceName="Channel3"/>
                <LogicalChannelNumber="3"/>
            </Service>
            <Service>
                <TextualID ServiceName="Channel5"/>
                <LogicalChannelNumber="6"/>
            </Service>
            <PackageAvailability>
                <CountryCode Availability="true">UK</CountryCode>
                <Cell>Scotland</Cell>
                <Cell>Ireland</Cell>
            </PackageAvailability>
        </Package>
        <Package Id="2">
            <PackageName Language="ENG">Provider1 Bouquet2</PackageName>
            <Service>
                <TextualID ServiceName="Channel7"/>
                <LogicalChannelNumber="2"/>
            </Service>
            <Service>
                <TextualID ServiceName="Channel8"/>
                <LogicalChannelNumber="4"/>
            </Service>
            <Service>
                <TextualID ServiceName="Channel9"/>
                <LogicalChannelNumber="5"/>
            </Service>
            <PackageAvailability>
                <CountryCode Availability="false">UK</CountryCode>
                <Cell>Scotland</Cell>
                <Cell>Wales</Cell>
            </PackageAvailability>
        </Package>
    </PackageDiscovery>
```

```
</ServiceDiscovery>
```

As an example, the "Broadcast" file below corresponds to the "Provider1". We retrieve in the broadcast discovery record the ServiceName from the package discovery record. The package record provides the logical channel number, while the broadcast record provides the complete information on the service.

	Provider1
Channels	Channel 2
	Channel 3
	Channel 5
	Channel 7
	Channel 8
	Channel 9

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <BroadcastDiscovery DomainName="provider1.com" Version="0">
        <ServiceList>
            <ServicesDescriptionLocation>
                <DescriptionLocation>bcg1</DescriptionLocation>
                <DescriptionLocation preferred="true">bcg2</DescriptionLocation>
            </ServicesDescriptionLocation>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.111.1.12" Port="8208" Source="192.100.100.50"
                        Streaming="udp" />
                </ServiceLocation>
                <TextualIdentifier DomainName="provider1.com" ServiceName="Channel2"/>
                <DVBTriplet OrigNetId="0" ServiceId="5002" TSId="202"/>
                <MaxBitrate>4</MaxBitrate>
                <ServiceAvailability>
                    <CountryCode Availability="true">UK</CountryCode>
                    <Cell>Scotland</Cell>
                    <Cell>Ireland</Cell>
                </ServiceAvailability>
                <ServiceAvailability>
                    <CountryCode Availability="true">FR</CountryCode>
                </ServiceAvailability>
            </SingleService>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.111.1.13" Port="8208" Source="192.100.100.50"
                        Streaming="udp" />
                </ServiceLocation>
                <TextualIdentifier DomainName="provider1.com" ServiceName="Channel3"/>
                <DVBTriplet OrigNetId="0" ServiceId="5003" TSId="203"/>
                <MaxBitrate>4</MaxBitrate>
                <ServiceAvailability>
                    <CountryCode Availability="true">UK</CountryCode>
                    <Cell>Scotland</Cell>
                    <Cell>Ireland</Cell>
                </ServiceAvailability>
                <ServiceAvailability>
                    <CountryCode Availability="true">FR</CountryCode>
                </ServiceAvailability>
            </SingleService>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.111.1.15" Port="8208" Source="192.100.100.50"
                        Streaming="udp" />
                </ServiceLocation>
                <TextualIdentifier DomainName="provider1.com" ServiceName="Channel5"/>
                <DVBTriplet OrigNetId="0" ServiceId="5005" TSId="205"/>
                <MaxBitrate>4</MaxBitrate>
                <ServiceAvailability>
                    <CountryCode Availability="true">UK</CountryCode>
                    <Cell>Scotland</Cell>
                    <Cell>Ireland</Cell>
                </ServiceAvailability>
                <ServiceAvailability>
                    <CountryCode Availability="true">FR</CountryCode>
                </ServiceAvailability>
            </SingleService>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.111.1.27" Port="8208" Source="192.100.100.50"
                        Streaming="udp" />
                </ServiceLocation>
                <TextualIdentifier DomainName="provider1.com" ServiceName="Channel7"/>
                <DVBTriplet OrigNetId="0" ServiceId="5007" TSId="207"/>
                <MaxBitrate>4</MaxBitrate>
                <ServiceAvailability>
                    <CountryCode Availability="false">UK</CountryCode>
                    <Cell>Scotland</Cell>
                    <Cell>Wales</Cell>
                </ServiceAvailability>
            </SingleService>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.111.1.28" Port="8208" Source="192.100.100.50"
                        Streaming="udp" />
                </ServiceLocation>
                <TextualIdentifier DomainName="provider1.com" ServiceName="Channel8"/>
                <DVBTriplet OrigNetId="0" ServiceId="5008" TSId="208"/>
                <MaxBitrate>4</MaxBitrate>
                <ServiceAvailability>
                    <CountryCode Availability="false">UK</CountryCode>
                    <Cell>Scotland</Cell>
                    <Cell>Wales</Cell>
                </ServiceAvailability>
            </SingleService>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.111.1.29" Port="8208" Source="192.100.100.50"
                        Streaming="udp" />
                </ServiceLocation>
                <TextualIdentifier DomainName="provider1.com" ServiceName="Channel9"/>
                <DVBTriplet OrigNetId="0" ServiceId="5009" TSId="209"/>
                <MaxBitrate>4</MaxBitrate>
                <ServiceAvailability>
                    <CountryCode Availability="false">UK</CountryCode>
                    <Cell>Scotland</Cell>
                    <Cell>Wales</Cell>
                </ServiceAvailability>
            </SingleService>
        </ServiceList>
    </BroadcastDiscovery>
</ServiceDiscovery>
```

NOTE: The Service Availability in the broadcast record matches the Package Availability in the package record, but can also contain more parameters, as shown with the first package and the first 3 services defined there.

6.4.3 Package and Broadcast Discovery with FEC

As an example, the " Package " file below corresponds to the "Provider2". For this service provider, only one bouquet is proposed: "Provider2 Bouquet". The bouquet "Provider2 Bouquet" contains the channels "Channel 15", "Channel 16", "Channel 17" and "Channel 18". The provider uses RTP streaming and provides FEC stream protection.

	Provider2 Bouquet
Channels	Channel 15
	Channel 16
	Channel 17
	Channel 18

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <PackageDiscovery DomainName="provider2.com" Version="0">
        <Package Id="3">
            <PackageName Language="ENG">Provider2 Bouquet</PackageName>
            <Service>
                <TextualID ServiceName="Channel 15"/>
            </Service>
            <Service>
                <TextualID ServiceName="Channel 16"/>
            </Service>
            <Service>
                <TextualID ServiceName="Channel 17"/>
            </Service>
            <Service>
                <TextualID ServiceName="Channel 18"/>
            </Service>
        </Package>
    </PackageDiscovery>
</ServiceDiscovery>
```

As an example, the "Broadcast" file below corresponds to the "Provider2".

	Provider2
Channels	Channel 15
	Channel 16
	Channel 17
	Channel 18

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <BroadcastDiscovery DomainName="provider2.com" Version="0">
        <ServiceList>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.222.2.15" Port="8208" Source="192.100.100.20"
                        Streaming="rtp" FECMaxBlockSizePackets="8" FECMaxBlockSizeTime="100"
                        FECOTI="MDBjMTA1MDE=">
                        <FECBaseLayer Address="224.222.2.15 Port="8210" Source="192.100.100.20" />
                        <FECEnhancementLayer Address="224.222.3.15 Port="4304"
                                             Source="192.100.100.20" />
                    </IPMulticastAddress>
                </ServiceLocation>
                <TextualIdentifier DomainName="provider2.com" ServiceName="Channel 15"/>
                <DVBTriplet OrigNetId="0" ServiceId="6001" TSId="5"/>
                <MaxBitrate>4</MaxBitrate>
            </SingleService>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.222.2.16" Port="8208" Source="192.100.100.20"
                        Streaming="rtp" FECMaxBlockSizePackets="8" FECMaxBlockSizeTime="100"
                        FECOTI="MDBjMTA1MDE=">
                        <FECBaseLayer Address="224.222.2.16 Port="8210" Source="192.100.100.20" />
                        <FECEnhancementLayer Address="224.222.3.16 Port="4304"
                                              Source="192.100.100.20" />
                    </IPMulticastAddress>
                </ServiceLocation>
                <TextualIdentifier DomainName="provider2.com" ServiceName="Channel 16"/>
                <DVBTriplet OrigNetId="0" ServiceId="6002" TSId="6"/>
                <MaxBitrate>4</MaxBitrate>
            </SingleService>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.222.2.17" Port="8208" Source="192.100.100.20"
                        Streaming="rtp" FECMaxBlockSizePackets="8" FECMaxBlockSizeTime="100">
                        <FECBaseLayer Address="224.222.2.17 Port="8210" Source="192.100.100.20" />
                    </IPMulticastAddress>
                </ServiceLocation>
                <TextualIdentifier DomainName="provider2.com" ServiceName="Channel 17"/>
                <DVBTriplet OrigNetId="0" ServiceId="6003" TSId="7"/>
                <MaxBitrate>4</MaxBitrate>
            </SingleService>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.222.2.18" Port="8208" Source="192.100.100.20"
                        Streaming="rtp" FECMaxBlockSizePackets="8" FECMaxBlockSizeTime="100">
                        <FECBaseLayer Address="224.222.2.18 Port="8210" Source="192.100.100.20" />
                    </IPMulticastAddress>
                </ServiceLocation>
                <TextualIdentifier DomainName="provider2.com" ServiceName="Channel 18"/>
                <DVBTriplet OrigNetId="0" ServiceId="6004" TSId="8"/>
                <MaxBitrate>4</MaxBitrate>
            </SingleService>
        </ServiceList>
    </BroadcastDiscovery>
</ServiceDiscovery>
```

NOTE: FEC Base Layer is compliant to SMPTE 2022-1 [7], so it is strongly recommended that the `Address` field is identical to the `Address` field of the content itself, and the `Port` field is +2 from the `Port` field of the content. Note that any other values will break compliance with SMPTE 2022-1 [7].

6.5 More Complex Examples for SD&S

This clause intends to present SD&S examples dealing with all possibilities offered by the DVB-IP Handbook.

6.5.1 Service Provider Discovery

6.5.1.1 Service Provider Discovery with Redundant Push/Pull Locations

The aim of the following record is to advertise a broadcast discovery record several times, pointing to different servers/multicast addresses. This allows the HNED to connect to different servers in case some of them are momentarily not responding.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <ServiceProviderDiscovery>
        <ServiceProvider DomainName="provider1.com" LogoURI="0" Version="0">
            <Name Language="ENG">Provider1</Name>
            <Description Language="ENG">Provider1 ADSL TV Offer</Description>
// one same package offering announced several times
            <Offering>
                <Pull Location="packages.provider1.com/dvb/sdns/">
                    <PayloadId Id="5">
                        <Segment ID="12cf" Version="46"/>
                        <Segment ID="30d2" Version="172"/>
                        <Segment ID="12" Version="2"/>
                    </PayloadId>
                </Pull>
                <Pull Location="packages.otherlocation.provider1.com/dvb/sdns/">
                    <PayloadId Id="5">
                        <Segment ID="12cf" Version="46"/>
                        <Segment ID="30d2"/>
                        <Segment ID="12"/>
                    </PayloadId>
                </Pull>
                <Push Address="224.1.1.5" Port="1234" Source="192.100.100.70">
                    <PayloadId Id="5"/>
                </Push>
                <Push Address="224.1.3.5" Port="5678" Source="192.100.100.70">
                </Push>
                <Push Address="224.1.7.5" Port="1234" Source="192.100.100.70">
                    <PayloadId Id="5"/>
                        <Segment ID="12cf" Version="46"/>
                        <Segment ID="30d2"/>
                        <Segment ID="12"/>
                    </PayloadId>
                </Push>
            </Offering>
// one broadcast offering announced
            <Offering>
                <Push Address="224.1.1.2" Port="1234" Source="192.100.100.70">
                    <PayloadId Id="2">
                        <Segment ID="12cf" Version="46"/>
                    </PayloadId>
                </Push>
            </Offering>
        </ServiceProvider>
    </ServiceProviderDiscovery>
</ServiceDiscovery>
```

In this example, the first offer is a package discovery record which is announced through 4 different possibilities, 2 pull and 3 push. Note that:

- For the pull mode, the location of the server is different, but segments are the same since the same content is available;
- For the push mode, it is not mandated to announce segments; it is not even mandated to announce the payload id. In that case, the HNED will check when receiving the header of the DVBSTP packets.

The second offer is a broadcast one; it has the same segment and version as the first offer, which is acceptable because we talk here about a different offer.

NOTE: The Payload@ID attribute is expressed in the XML data structure in hexadecimal coded with 1 or 2 characters, while it is coded with exactly 2 hexadecimal characters in the URL of the HTTP request.
The Segment@ID attribute is expressed in the XML data structure in hexadecimal coded with 1 to 4 characters, while it is coded with exactly 4 hexadecimal characters in the URL of the HTTP request.
The Segment@Version attribute is expressed in the XML data structure in decimal, while it is coded with exactly 2 hexadecimal characters in the URL of the HTTP request.
For example, for the first pull location :

```
<Pull Location="packages.provider1.com/dvb/sdns/">
   <PayloadId Id="5">
      <Segment ID="12cf" Version="46"/>
      <Segment ID="30d2" Version="172"/>
      <Segment ID="12" Version="2"/>
   </PayloadId>
</Pull>
```

the HTTP requests to retrieve the segments are :

```
GET /dvb/sdns/service_discovery?id=provider1.com&Payload=05&Segment=12cf&Version=2e
GET /dvb/sdns/service_discovery?id=provider1.com&Payload=05&Segment=30d2&Version=ac
GET /dvb/sdns/service_discovery?id=provider1.com&Payload=05&Segment=0012&Version=02
```

6.5.1.2 Service Provider Discovery with Complementary Push/Pull Locations

In the previous example we talked about redundancy for the announcement of the SD&S data. It consumes server resource and bandwidth resource to do that, while a service provider may want to optimize the resources to send the SD&S data.

Let is base the following example on a broadcast offering by a service provider (other payload ids can of course be used). We assume a service provider has 200 TV channels. If no splitting at all was performed, the offering can be included in a unique segment sent on one multicast group. Assuming an average size of each single service XML record of 1 k-byte, we have 200 k-bytes to send in a maximum delay of 30 seconds. This produces a bitrate of 53 kbits/s in the SD&S multicast.

At the opposite, we can build 200 segments, each one containing the description for only one service, and assign a different multicast group to the sending of each segment such as illustrated below. There is also the possibility to split the HTTP server load, here 2 servers are defined, to hold odd and even segment numbers (but any split is possible, with any number of servers).

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <ServiceProviderDiscovery>
        <ServiceProvider DomainName="provider1.com" LogoURI="0" Version="0">
            <Name Language="ENG">Provider1</Name>
            <Description Language="ENG">Provider1 ADSL TV Offer</Description>
// one broadcast offering announced in several pieces
            <Offering>
                <Push Address="224.1.7.0" Port="1234" Source="192.100.100.70">
                    <PayloadId Id="2"/><Segment ID="0"/></PayloadId>
                </Push>
                <Push Address="224.1.7.1" Port="1234" Source="192.100.100.70">
                    <PayloadId Id="2"/><Segment ID="1"/></PayloadId>
                </Push>
                <Push Address="224.1.7.2" Port="1234" Source="192.100.100.70">
                    <PayloadId Id="2"/><Segment ID="2"/></PayloadId>
                </Push>
                [..................................................................]
                <Push Address="224.1.7.198" Port="1234" Source="192.100.100.70">
                    <PayloadId Id="2"/><Segment ID="c6"/></PayloadId>
                </Push>
                <Push Address="224.1.7.199" Port="1234" Source="192.100.100.70">
                    <PayloadId Id="2"/><Segment ID="c7"/></PayloadId>
                </Push>
                <Pull Location="services0to99.provider1.com/dvb/sdns/">
                    <PayloadId Id="2">
                        <Segment ID="0"/>
                        <Segment ID="2"/>
                        <Segment ID="4"/>
                        [...................]
                        <Segment ID="c4"/>
                        <Segment ID="c6"/>
                    </PayloadId>
                </Pull>
                <Pull Location="services100to199.provider1.com/dvb/sdns/">
                    <PayloadId Id="2">
                        <Segment ID="1"/>
                        <Segment ID="3"/>
                        <Segment ID="5"/>
                        [...................]
                        <Segment ID="c5"/>
                        <Segment ID="c7"/>
                    </PayloadId>
                </Pull>
            </Offering>
        </ServiceProvider>
    </ServiceProviderDiscovery>
</ServiceDiscovery>
```

Sending each multicast stream with a cycle time of 30 seconds makes a bitrate of only 266 bits/s for each. So the HNED which wants to check only a few channel descriptions generates a bandwidth of only a few times these 266 bits/s.

If the HNED wants to monitor everything simultaneously, it will join all the multicast groups and receive a total bitrate of 52 kbits/s - the same as if only one stream was used. This can be used for example at boot time to acquire the complete service plan. Since each stream cycles in 30 seconds, the HNED still receives the complete information in 30 seconds.

Then since SD&S updates are not frequent, the HNED might decide to listen to only one multicast group at a time. This uses a permanent bandwidth of only 266 bits/s. Each segment is received in (maximum) 30 seconds. So at most after 30 seconds, the HNED leaves the current group and joins the next one. The total time to scan all the information is now (maximum) 1 hour and 40 minutes.

Between the 2 extreme options - all in one segment and one service per segment with one stream per segment - the service provider has much flexibility to find a good compromise between the use of bandwidth, the time it takes to check updates and the number of multicast addresses used (which may be limited).

6.5.1.3 Simplest Service Provider Discovery Offer

Since the Payload element is not required for push location, the following XML table is the simplest possible one for service provider discovery. Of course the only way for the HNED to know what is offered is to join the multicast groups and check the payload id within the DVBSTP headers. Note that the Source field of the Push element is optional, it was not set here.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <ServiceProviderDiscovery>
        <ServiceProvider DomainName="provider1.com" LogoURI="0" Version="0">
            <Name Language="ENG">Provider1</Name>
            <Description Language="ENG">Provider1 ADSL TV Offer</Description>
            <Offering>
                <Push Address="224.1.1.5" Port="1234" />
            </Offering>
            <Offering>
                <Push Address="224.1.1.2" Port="1234" />
            </Offering>
        </ServiceProvider>
    </ServiceProviderDiscovery>
```

```
</ServiceDiscovery>
```

6.5.2 Broadcast Offering with Multiple Multicast/RTSP Locations

It is also possible to provide several locations that allow access to the content. The following example presents a complex example with several push and pull locations. There are 2 RTSP servers able to provide session management for this live content, one multicast stream using UDP streaming without FEC, and one multicast stream using RTP with FEC.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <BroadcastDiscovery DomainName="provider1.com" Version="0">
        <ServiceList>
            <SingleService>
                <ServiceLocation>
                    <RTSPURL>rtsp://live.provider1.com/Channel12.mpg</RTSPURL>
                    <IPMulticastAddress Address="224.111.1.12" Port="8208" Source="192.100.100.50"
                        Streaming="udp" />
                    <IPMulticastAddress Address="224.111.1.22" Port="4302"
                        Streaming="rtp" FECMaxBlockSizePackets="8" FECMaxBlockSizeTime="100">
                        <FECBaseLayer Address="224.111.1.22 Port="4304" Source="192.100.100.50" />
                    </IPMulticastAddress>
                    <RTSPURL>rtsp://live.proxy.provider1.com/Channel12.mpg</RTSPURL>
                </ServiceLocation>
                <TextualIdentifier DomainName="provider1.com" ServiceName="Channel2"/>
                <DVBTriplet OrigNetId="0" ServiceId="5002" TSId="202"/>
                <MaxBitrate>4</MaxBitrate>
            </SingleService>
        </ServiceList>
    </BroadcastDiscovery>
</ServiceDiscovery>
```

NOTE: The order of service locations is not important, therefore there is no preference or default location implied.

6.5.3 Single Big Push Discovery

Since DVBSTP has all information in the header of the packet to discriminate payload ids and segments, it is possible for a service provider to provide all the SD&S data on one single multicast address. The following XML tables are an example of such a system.

The entry point provides the multicast address 224.1.1.1 to discover the Service Provider Discovery record. This record is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <ServiceProviderDiscovery>
        <ServiceProvider DomainName="provider1.com" LogoURI="0" Version="0">
            <Name Language="ENG">Provider1</Name>
            <Description Language="ENG">Provider1 ADSL TV Offer</Description>
            <Offering>
                <Push Address="224.1.1.1" Port="1234" />
            </Offering>
            <Offering>
                <Push Address="224.1.1.1" Port="1234" />
            </Offering>
        </ServiceProvider>
    </ServiceProviderDiscovery>
</ServiceDiscovery>
```

Let is say that the first offering is a package record, and the second one is a broadcast record.

The multicast group 224.1.1.1 is carrying three different payload ids data: the service provider discovery record (1), the broadcast record (2) and the package record (5). Each payload id can have several segments.

The HNED will perform the parsing of payload ids and segment thanks to the DVBSTP header.

6.5.4 Multiple Coding Formats

The SD&S data structure can provide audio and video formats used by the service. When a service provider is able to present the same service using several different formats (size, coding, etc.), it has to advertise several services in the service list.

The following example shows the case of one service being available in SD and HD formats.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <BroadcastDiscovery DomainName="provider1.com" Version="0">
        <ServiceList>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.111.1.12" Port="8208" Source="192.100.100.50"
                        Streaming="udp" />
                </ServiceLocation>
                <TextualIdentifier DomainName="provider1.com" ServiceName="Channel-1 SD"/>
                <DVBTriplet OrigNetId="0" ServiceId="5002" TSId="202"/>
                <MaxBitrate>4</MaxBitrate>
                <SI ServiceType="" PrimarySISource="XML">
                    <Name Language="ENG">Channel 1 - SD</Name>
                    <Name Language="FRA">Canal 1 - SD</Name>
                    <Description Language="ENG">This is the channel 1 in SD</Description>
                    <Description Language="FRA">Ceci est le canal 1 en SD</Description>
                </SI>
                <AudioAttributes>
                    <Coding href="urn:mpeg:mpeg7:cs:AudioCodingFormatCS:2001:3.2">
                        <Name>MPEG-1 Audio Layer II</Name>
                    </Coding>
                    <NumOfChannels>2</NumOfChannels>
                </AudioAttributes>
                <VideoAttributes>
                    <Coding href="urn:mpeg:mpeg7:cs:VisualCodingFormatCS:2001:2.2.2">
                        <Name>MPEG-2 Video Main Profile @ Main Level</Name>
                    </Coding>
                </VideoAttributes>
            </SingleService>
            <SingleService>
                <ServiceLocation>
                    <IPMulticastAddress Address="224.111.1.13" Port="8208" Source="192.100.100.50"
                        Streaming="udp" />
                </ServiceLocation>
                <TextualIdentifier DomainName="provider1.com" ServiceName="Channel-1 HD"/>
                <DVBTriplet OrigNetId="0" ServiceId="5002" TSId="203"/>
                <MaxBitrate>10</MaxBitrate>
                <SI ServiceType="" PrimarySISource="XML">
                    <Name Language="ENG">Channel 1 - HD</Name>
                    <Name Language="FRA">Canal 1 - HD</Name>
                    <Description Language="ENG">This is the channel 1 in HD</Description>
                    <Description Language="FRA">Ceci est le canal 1 en HD</Description>
                </SI>
                <AudioAttributes>
                    <Coding href="urn:dvb:ipdc:esg:cs:AudioCodecCS:5.10.2">
                        <Name>MPEG-4 High Efficency Advanced Audio Profile @ Level 2</Name>
                    </Coding>
                    <NumOfChannels>2</NumOfChannels>
                </AudioAttributes>
                <VideoAttributes>
                    <Coding href="urn:dvb:ipdc:esg:cs:VideoCodecCS:9.4.12">
                        <Name>H264 High Profile @ Level 4.0</Name>
                    </Coding>
                </VideoAttributes>
            </SingleService>
        </ServiceList>
    </BroadcastDiscovery>
</ServiceDiscovery>
```

6.6 Regionalization and Logical Channel Numbers

A HNED has used the SD&S mechanism to discover the available service providers and services in the IPTV network. The service provider assigns Logical Channel Numbers (LCN) to services described in the SD&S records. The LCN defines the service provider's preferred ordering of the available services in the HNED channel list. The LCN is a number associated with every service, allowing the presentation of the service and its selection.

In this example all regional Channel2 services are listed contiguously, but they all use the same channel number since they are not supposed to exist simultaneously.

LCN	Channel	Regional Availability (Cell 1)	Regional Availability (Cell 2)
1	Channel2 region1	**true**	false
1	Channel2 region2	false	**true**
1	Channel2 region3	false	false
1	Channel2 region4	false	false
2	Channel3	**true**	**true**
3	Channel5	**true**	**true**

In Cell 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <PackageDiscovery DomainName="provider1.com" Version="0">
        <Package Id="1">
            <PackageName Language="ENG">Provider1 Bouquet1</PackageName>
            <Service>
                <TextualID ServiceName="Channel2 region1"/>
                <LogicalChannelNumber=1/>
            </Service>
            <Service>
                <TextualID ServiceName="Channel3"/>
                <LogicalChannelNumber=2/>
            </Service>
            <Service>
                <TextualID ServiceName="Channel5"/>
                <LogicalChannelNumber=3/>
            </Service>
        </Package>
    </PackageDiscovery>
</ServiceDiscovery>
```

In Cell 2:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <PackageDiscovery DomainName="provider1.com" Version="0">
        <Package Id="1">
            <PackageName Language="ENG">Provider1 Bouquet1</PackageName>
            <Service>
                <TextualID ServiceName="Channel2 region2"/>
                <LogicalChannelNumber=1/>
            </Service>
            <Service>
                <TextualID ServiceName="Channel3"/>
                <LogicalChannelNumber=2/>
            </Service>
            <Service>
                <TextualID ServiceName="Channel5"/>
                <LogicalChannelNumber=3/>
            </Service>
        </Package>
    </PackageDiscovery>
</ServiceDiscovery>
```

7 Connection to the Live Service

At this step, the HNED has collected the XML file that contains the BroadcastDiscovery structure. The HNED can get in this file the information to access the different channels composing its bouquet.

7.1 Connection possibilities

A Live TV service may be accessed by an individual HNED in the following ways:

- Using **IGMP** (Internet Group Management Protocol):
  In this case, the HNED has collected a Multicast IP address for this Live TV service. To display this Live TV channel, the HNED sends an IGMP Report request to this Multicast IP address in order to subscribe to this multicast group. Multicast Content Services use IGMP version 3 with Source Specific Multicast. This allows significant scalability and implementers should note that the previous version of IGMP is not allowed (see next clause for details).

- Using **RTSP** (Real Time Streaming Protocol):
  In this case, the element "Service Location" in the service discovery record signals the use of RTSP and gives all the information necessary to issue the appropriate RTSP method. Parameters required for the IGMP message will be acquired via the SETUP method from RTSP.

For an example of a "Package" file and a "Broadcast" file, see clause 6.3.

7.1.1 Multicast Connection

The DVB-IP Handbook mandates IGMPv3 (RFC 3376 [15]) for the IPI-1 interface. What does it mean?

7.1.1.1 IGMPv1

IGMPv1 (RFC 1112 [16]) defines the following message:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Type  |    Unused     |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Group Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The 2 messages types (`Type`) are:

- Host Membership Query, value=1
- Host Membership Report, value=2

As version is 1, this gives for the first octet:

- Host Membership Query, value=0x11
- Host Membership Report, value=0x12

The `Unused` field is set to zeroes.

7.1.1.2 IGMPv2

IGMPv2 defines the following message:

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     | Max Resp Time |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Group Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The message `Types` values are:

- 0x11: Membership Query (General Query or Specific Query).
- 0x16: v2 Membership Report. The IP packet is sent to eh specific multicast address.
- 0x17: Leave Group. The IP packet is sent to the all routers group (224.0.0.2).

The `Max Resp Time` field reflects the maximum time before sending a response for the host; this allows managing timers in a more efficient way.

7.1.1.3 IGMPv3

IGMPv3 defines the following Membership Query message:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type = 0x11  | Max Resp Code |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Group Address                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Resv  |S| QRV |     QQIC      |     Number of Sources (N)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address [1]                      |
+-                                                             -+
|                       Source Address [2]                      |
+-                              .                              -+
.                               .                               .
.                               .                               .
+-                                                             -+
|                       Source Address [N]                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The `Max Resp Code` field is quite equivalent to the v2 `Max Resp Time`, with possibilities for more complex values.

7.1.1.4 Impact is on the HNED

In IGMPv3 (RFC 3376 [15]), clause 7.2.1 says: "In order to be compatible with older version routers, IGMPv3 hosts MUST operate in version 1 and version 2 compatibility modes". So having a DVB-IP compliant HNED means that it follows the IGMPv3 spec, meaning that it must conform to this statement.

The detection of which version of IGMP runs on the router is done by looking at the Query message received:

- IGMPv1 Query: length = 8 octets AND Max Resp Code field is zero.
- IGMPv2 Query: length = 8 octets AND Max Resp Code field is non-zero.
- IGMPv3 Query: length >= 12 octets.

Thus it is possible that the HNED be connected to network with previous versions of IGMP, though such a configuration would not take advantage of IGMPv3 enhancement.

7.2 Transport of the stream

The video content is streamed using an MPEG-2 transport stream, as defined in TS 101 154 [2], which is then encapsulated in RTP/UDP or directly in UDP. Usually a Single Program Transport Stream (SPTS) is used as only the bandwidth for the selected content is needed. However Multi Program Transport Streams (MPTS) are not out ruled.

The information if RTP/UDP or UDP encapsulation is used is provided by the SD&S Broadcast Discovery record attribute IPMulticastAddress@Streaming or the BCG locator for multicast services and by the RTSP transport header for unicast services:

- A IPMulticastAddress@Streaming value of "rtp" or a BCG locator with the syntax "rtp://…." indicates RTP/UDP encapsulation.
- A IPMulticastAddress@Streaming value of "udp"or a BCG locator with the syntax "udp://…." indicates direct UDP encapsulation.

In case the IPMulticastAddress@Streaming attribute is not defined in the SD&S record RTP/UDP encapsulation is assumed.

A RTSP transport header of "RTP/AVP/UDP" indicates RTP/UDP encapsulation. A RTSP transport header of either "MP2T/H2221/UDP" or "RAW/RAW/UDP" indicates direct UDP encapsulation.

8 Network Management and Provisioning

Network provisioning is optional in the specification, but for those implementers that wish to use it, this clause gives an overview and help in constructing both the client and server side of the system.

8.1 Overview

One of the commercial aims of the specification is for the HNED to be bought by a person in a retail store, bring it home, plug it in and then be able to quickly view entertainment. The technical group believed that one way to meet this commercial requirement was to automatically provision the HNED using remote servers and back-office systems.

8.1.1 What is Network Provisioning Meant to Solve?

If we look at the specifics of the commercial aim and translate them into steps the user would do manually, we get a flow something like:

1) Customer buys HNED in local electronics store.

2) Customer connects the HNED to their home DNG and powers the HNED on.

3) The unit starts up with a default configuration and gets onto the Service Provider network.

4) Customer then tells Service Provider the type of box and other information chooses services that they want, billing etc. Verified by billing server, etc.

5) The new services require a new configuration, perhaps even new software.

When you look at this technically this is significantly more complex because the network needs to detect a new box, e.g. take an inventory, provision and reprovision the HNED and then monitor the HNED for any problems. This is what Network Provisioning and Management was specifically designed to provide.

8.1.2 What does a Service Provider Management System Need to Do?

If we look inside a typical Service Provider Management System you get something as in figure 12.

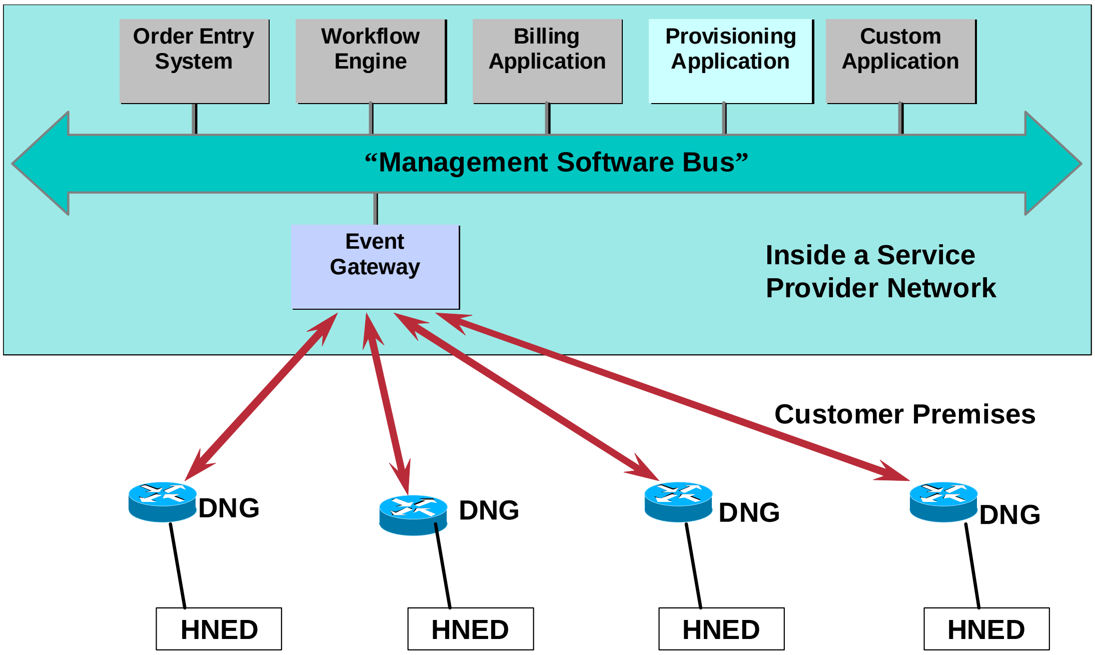

Figure 12: Service Provider Management System overview

A service provider management system has many systems for order entry, billing and provisioning all linked together in some way perhaps by a "Management Software Bus" so that when a customer orders a new service it is reflected in all internal systems. In the DVB-IP specification case, we assume that the HNED provisioning application is linked to these systems so that when a customer chooses a new service, it is reflected in a new configuration made by the "Provisioning Application" which then pushes it to the DNG via the "Event Gateway."

The Event Gateway is the umbilical link between the HNED and the Network Provisioning and Management. It is designed to be scalable to millions of subscribers using standard Internet XML over HTTP(S) technology, where similar techniques have been used successfully. The only component that the Service Providers needs additionally to network provision and manage the HNEDs is that box. The specification of the Event Gateway is not included in the DVB-IP specification; however, the protocol defines what is expected in replies to the HNED.

8.2 Key Components

The key components of the system are the Event Gateway, the Event Agent, Configuration Agent and Inventory Agent as shown in the figure 13.

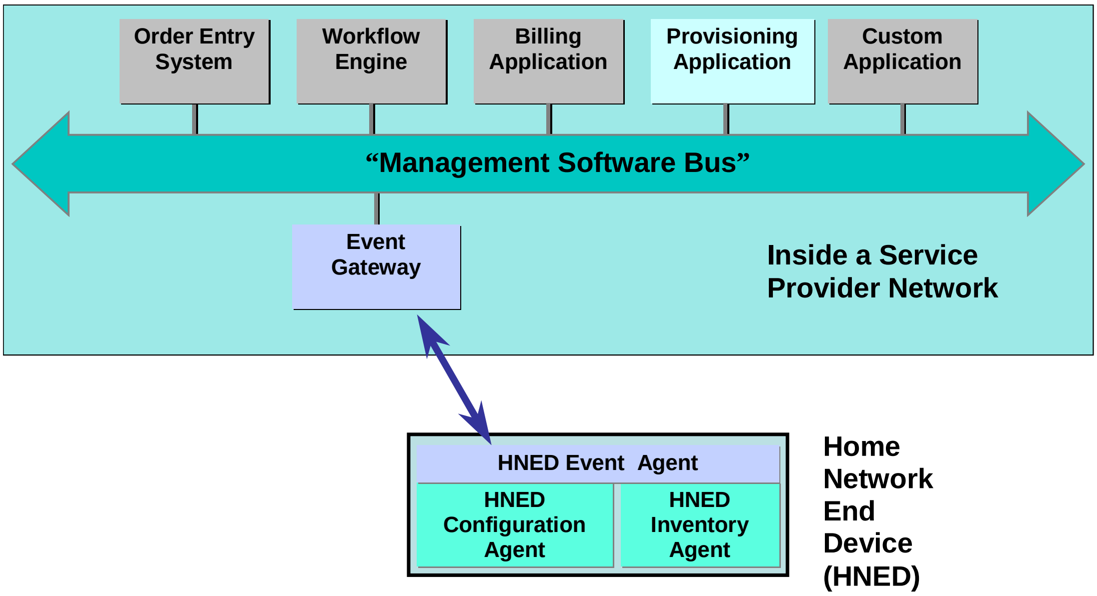

Figure 13: Management System components

8.2.1 Event Gateway

The Event Gateway lies in the Service Provider network and needs to:

- Send configurations.
- Send events.
- Receive inventories.
- Receive configurations.
- Receive status.
- Receive and react to events.

8.2.2 Event Agent

The Event Agent manages communications with the Event Gateway. It performs the functions:

- Ensures reliable message delivery.
- Encrypts the messages (SSL).
- Provides an Event API for other agents.
- Delivers received events to the appropriate agent.
- Forwards other agent events to the Event Gateway.
- Fails over to a secondary Event Gateway if it detects the gateway or link to be down.

8.2.3 Configuration Agent

The Configuration Agent manages the network configuration of HNED. It performs the functions:

- Receives configuration commands from the Event Agent.
- Applies the configuration commands to the HNED.

- Synchronizes configuration changes.
- Encrypts the configuration (SSL).
- Reports configuration on request.
- Reports configuration changes as they occur.

8.2.4 Inventory Agent

The Inventory Agent manages the list of components and add-ins of the HNED, for example, software version number, plug-in cards etc. It performs the functions:

- Reports inventory on request.
- Reports inventory changes when detected.
- Reports status on request.
- Reports status changes when detected.

8.3 DHCP Address to Configuration Flow

The whole boot sequence, from start-up to service discovery, is described in clause 5, but as network provisioning is optional, we will describe what happens if network provisioning is used here.

1) HNED is powered on and any internal testing performed.

2) HNED uses DHCP to obtain an IP address and other information.

3) The DHCP server returns a valid IP address in the DHCP next server "*siaddr"* field which means that there is a network provisioning server available. The "*siaddr*" is the IP address of the provisioning server.

4) An HTTP GET "*boot*" event should then be sent with the appropriate manufacture's name, HNED flash (read-only) and RAM (read-write) storage.

5) The Provisioning Server return a 200 series success status to which assures that the provisioning server is operational and will provision this box. If the status is not returned then (4) is tried according to the congestion avoidance mechanism.

6) The Provisioning Server fills in the XML sent with the GET. The key field being the response action:

   - Response action is "none" then the HNED will send a GET "*configure*" event to the provisioning server and begin regular event polling.
   - Response action field is "inventory", "status", "configure", "update" or "boot" then the appropriate actions are performed until the response action is "none" When "none" happens then regular event polling begins.

Regular event polling consists of sending an HTTP GET "*event*" with a time as set in the "*interval*" field of the Configure XML DTD. The events will be processed until the response action field is "*none"* whereupon a new polling interval is started.

8.4 Worked Example

We will now take the commercial example from clause 8.1.1 and turn it into a worked example of what happens from a technical viewpoint.

1) Customer buys HNED in local electronics store.

2) Customer connects the HNED to their home DNG (Delivery Network Gateway) and powers the HNED on.

3) HNED sends out the standard DHCP DISCOVER message with DVB mandatory DHCP options.

4) DHCP Server returns HNED IP Address and other options, and the Event Gateway IP address in *siaddr.*

5) HNED sends an HTTP GET to /dvb/boot using the Event Gateway's IP address.

6) The Event Gateway receives the message and either it or the Provisioning/Billing Application detects this is a new HNED and user of video. It also uses the manufacturer's name and memory size to decide whether the HNED can be provisioned. We assume that the HNED will be able to be Network Provisioned.

   In this example, the Network Service Provider has made a policy that when a new HNED and user is found then the HNED will use special "initial" configuration. This "initial" configuration typically only allows the HNED to reach very limited services until completion of a form for billing and other customer information.

7) The Event Gateway needs an inventory so that it can send an appropriate initial configuration. It therefore sends the XML event return action as "*inventory*".

8) The HNED replies by sending an HTTP POST to /dvb/inventory with the inventory in XML.

9) After the congestion avoidance time and assuming POST success, the HNED sends an HTTP GET "*event*".

10) The Event Gateway replies with the special "initial" configuration, given to it from the provisioning application, with the XML event return event action = "update" and the configuration in the configure clause of the returned XML.

11) The HNED reconfigures.

    In this example, while parsing the XML the HNED recognizes that a form for billing and other customer information is required, and generates the form. The form or communication protocols with the back office systems in the Network Service Provider are not specified by Network Provisioning, though it is possible to send a URL or other pointer in the initial configuration XML. Once the customer has completed the form and the information verified, the Network Service Provider sends a "final" configuration that allows access to the sets of services.

    In this example, this final configuration also requires a change of IP address. This requires the help of the DHCP server which we assume is under the command of the Network Service Provider (see clause 5.2 on addressing).

12) First the HNED needs to obtain the new IP address and only then the new reconfiguration. The DHCP Server sends a FORCERENEW message to the HNED (RFC 3203 [12]) and NAKs the DHCP REQUEST.

13) The HNED returns to initial DHCP state and sends a standard DHCP DISCOVER message with the mandatory DHCP options. This is the same state as (3) above.

14) DHCP Server returns the new HNED IP Address and other options, and, in this example, the same Event Gateway IP address in *siaddr.*

15) The new IP address means that the HNED sends an HTTP GET to /dvb/boot using the Event Gateway's IP address.

16) The Event Gateway returns an "*update*" event action with the final configuration.

17) The HNED reconfigures to the final configuration.

18) The HNED sends an HTTP GET "*event*" and the Event Gateway replies with an XML event action of "*none*". Final configuration is complete.

19) The HNED now goes back to normal polling where the HNED periodically sends an HTTP GET to /dvb/event. The XML event returns "*none*" unless the Event Gateway wants some action.

9 Typical applications available within the scope of the DVB-IP phase 1 Handbook

DVB-IP Phase 1 is a significant step in standardizing entertainment video over IP home network; however, it does not cover all possibilities or areas for standardization. This clause attempts to outline its boundaries with the belief that future versions of the standard will extend the scope.

The boundaries can be broken down into a number of areas:

- Audio/Video transport and codecs.
- Topology.
- Networking Addressing and Discovery.
- Service Provisioning.
- Network Level Security.
- Operation over different physical networks and Quality of Service.
- DNG/HNED only networks.

9.1 Video transmission and codecs

The current version of the DVB-IP specification only addresses the use of an MPEG-2 transport stream for the delivery of content. It does not address separation of the transport stream into elementary streams or any other carrier other than the MPEG2 transport stream.

The transport over IP is via RTP and UDP or via UDP directly (without RTP). For the later the network has to ensure that no packet reordering occurs.

In case of RTP encapsulation a single PCR per MPTS should be used.

AL-FEC is only supported with RTP/UDP encapsulation. It is not supported for direct UDP encapsulation of the transport stream.

Supported media formats are given in TS 101 154 [2].

9.2 Topology

The current version of the DVB-IP specification is limited to the following simple in-home network topologies:

- DNG/HNED Only Networks.
- Single Segment Home Networks with single address space and single DHCP server.

These are quite restrictive topologies but simple enough to satisfy the majority of current uses cases. This means that a network consisting of two DNGs in the home must be on independent and unconnected network segments.

9.3 Networking Addressing and Discovery

The standard uses DHCP to obtain network addressing and several other pieces of information but the option table is deliberately short to make client implementation simple in the HNED. However the implementation does use the new server message outlined in RFC 3203 [12] "FORCERENEW" which is not usually implemented in most DHCP servers.

The DHCP message also requires a unique identifier so the reuse of MAC addresses by whatever method is not allowed.

The DHCP server non-availability has been designed to be an unusual occurrence so whilst the use of RFC 3927 [11] is recommended in emergency, temporary situations; a DHCP server will be required for a DVB-IP home network to function normally.

9.4 Service Provisioning

The service provisioning covers initial and some subsequent network level provisioning, however, it does not cover diagnostics.

9.5 Network Level Security

Network level security, for example, denial of service attacks are not covered in the specification.

9.6 Operation over different physical networks and Quality of Service

The design of DVB-IP Phase 1 is physical layer independent and relies on the IP network to provide the required quality of service. The DVB-IP specifications are easily met on most wired networks but less so by in-home wireless networks, particularly 802.11 networks.

The specification defines only the user priority classes for the different traffic types and the related DSCP and IEEE 802.1d [14]priority values. No QoS enforcement mechanisms are defined.

9.7 DNG/HNED Only Networks

The requirement for a combination DNG/HNED e.g. a DSL modem combined with a set-top box, means that phase 1 treats this box as effectively a DNG and is outside of the scope of the specification.

However, if this box has any Ethernet or other interface capable of providing a network for example IEEE 802.11a/g [13] wireless LAN then it falls under the specification of DVB-IP.

10 Discovery of BCG information

This Clause explains how an HNED can get access to Broadband Content Guide descriptions. BCG data are TV-Anytime content guide descriptions, which are available on a given always-on bidirectional IP network.

10.1 Discovery of BCG Providers

Available BCG providers are discovered through the BCGDiscovery records in the SD&S information, as shown in the following example.

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceDiscovery xmlns="urn:dvb:ipisdns:2006" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <BCGDiscovery DomainName="bcgprovider1.com">
        <BCG Id="bcg1">
            <Name Language="eng">Provider1 BCG</Name>
            <TransportMode>
                <HTTP Location="bcg.provider1.com/dvb/sdns/">
                    <PayloadId Id="a1">
                        <Segment ID="7b" Version="4"/>
                        <Segment ID="4d5" Version="17"/>
                        <Segment ID="1" Version="2"/>
                    </PayloadId>
                    <PayloadId Id="a2">
                        <Segment ID="0" Version="4"/>
                    </PayloadId>
                    <PayloadId Id="a3">
                        <Segment ID="0"/>
                    </PayloadId>
                    <PayloadId Id="a4">
                        <Segment ID="135" Version="71"/>
                    </PayloadId>
                    <PayloadId Id="a5">
                        <Segment ID="4"/>
                    </PayloadId>
                    <PayloadId Id="a6">
                        <Segment ID="66"/>
                    </PayloadId>
                    <PayloadId Id="a7">
                        <Segment ID="1"/>
                    </PayloadId>
                </HTTP>
                <DVBSTP Port="8207" Address="224.222.2.46"/>
                <HTTP Location="bcg.soap.provider1.com/dvb/sdns/" SOAP="true" />
            </TransportMode>
            <TargetProvider>sport-provider.com</TargetProvider>
        </BCG>
        <BCG Id="bcg2">
            <Name Language="eng">Provider2 BCG</Name>
            <TransportMode>
                <DVBSTP Port="5512" Address="224.235.32.4"/>
            </TransportMode>
            <TargetProvider>news-provider.com</TargetProvider>
        </BCG>
    </BCGDiscovery>
</ServiceDiscovery>
```

The previous example provides BCG discovery information for "Provider1" and "Provider2". The BCG1 provides content guide for content provider "sport-provider.com", and is transmitted over DVBSTP or HTTP. In the case of HTTP, all PayloadIds are identified, with their segment (and optionally their version). Furthermore, the BCG1 advertises a SOAP server. The BCG2 is only retrievable via DVBSTP.

If multiple BCG records are available then one may be specified as preferred in the ServicesDescriptionLocation, otherwise the choice is implementation dependent e.g. it may be based on user preference.

10.2 Access to BCG Information

Access to BCG Information from a BCG provider is done in Push or Pull mode, and optionally using SOAP Queries.

10.2.1 DVBSTP and HTTP Mechanisms

For Push mode and HTTP Pull mode, BCG data are made available to HNEDs as BiM-encoded TV-Anytime fragments encapsulated in containers, as specified in TS 102 323 [9]. They can be transported in Push mode over DVBSTP or in Pull mode over HTTP.

Below is an example of a BCG instance document received by the HNED.

```
<?xml version="1.0" encoding="UTF-8"?>
<tva:TVAMain xml:lang="eng" xmlns:tva="urn:tva:metadata:2005"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <tva:ProgramDescription>
      <tva:ProgramInformationTable>
         <tva:ProgramInformation programId="crid://cp/051103120018021612A">
            <tva:BasicDescription>
               <tva:Title>Program1</tva:Title>
               <tva:Synopsis length="short">This is the synopsis of the movie</tva:Synopsis>
               <tva:Genre href="urn:tva:metadata:cs:ContentCS:2005:3.4">
                  <tva:Name>Film</tva:Name>
               </tva:Genre>
               <tva:Duration>P0DT01H30M</tva:Duration>
            </tva:BasicDescription>
         </tva:ProgramInformation>
      </tva:ProgramInformationTable>
      <tva:ProgramLocationTable>
         <tva:Schedule serviceIDRef=" extreme-sport.sport-provider.com" start="2003-10-
21T00:00:00+00:00" end="2003-10-21T23:59:59+00:00">
            <tva:ScheduleEvent>
               <tva:Program crid="crid://cp/051103120018021612A"/>
               <tva:PublishedStartTime>2005-11-11T07:00:00Z</tva:PublishedStartTime>
               <tva:PublishedDuration>P0DT01H30M</tva:PublishedDuration>
            </tva:ScheduleEvent>
         </tva:Schedule>
      </tva:ProgramLocationTable>
      <tva:ServiceInformationTable>
         <tva:ServiceInformation serviceId=" extreme-sport.sport-provider.com">
            <tva:Name>Extreme Sport</tva:Name>
         </tva:ServiceInformation>
      </tva:ServiceInformationTable>
   </tva:ProgramDescription>
</tva:TVAMain>
```

The previous example provides schedule and content information on "Program1" which is broadcasted on "Extreme Sport".

Note that updated versions of BCG records can be detected by the terminal using the same mechanisms as for SD&S records, i.e. using the version field in DVBSTP header in the case of DVBSTP and the VersionNumber field in the URL for the http request in the case of HTTP.

10.2.2 SOAP Query Mechanism

The following clause defines some guidelines for the usage of the BCG SOAP query mechanism. This mechanism is used as defined by TV-Anytime in the system specification (TS 102 822-2 [4]) and more specifically in the bi-directional specification (TS 102 822-6-1 [5]). As such the ETSI specifications should be referred to for more detailed guidelines.

10.2.2.1 Typical Flow Of Events

In order to use the SOAP query mechanism the following steps outline a typical flow of events from SD&S record to BCG acquisition:

- Acquire BCG record(s) via SD&S.
- Use the HTTP@Location URL, where the HTTP@SOAP is "true", to discover the location of a BCG provider.
- Send a describe_get_Data SOAP request to the BCG provider.
- Check response to ensure provider has required capabilities.
- Send a get_Data SOAP request to the BCG provider.

If the capabilities of the BCG providers are not suitable (e.g. a required table is not available) then a describe_get_Data may be performed on the next BCG provider until a suitable provider is found. It may be that the data required might be stored across a range of BCG providers in which case multiple queries to multiple providers may be required to satisfy a particular request.

10.2.2.2 Protocol stack

Figure 14 outlines the protocols required to deliver a BCG using the SOAP Query mechanism.

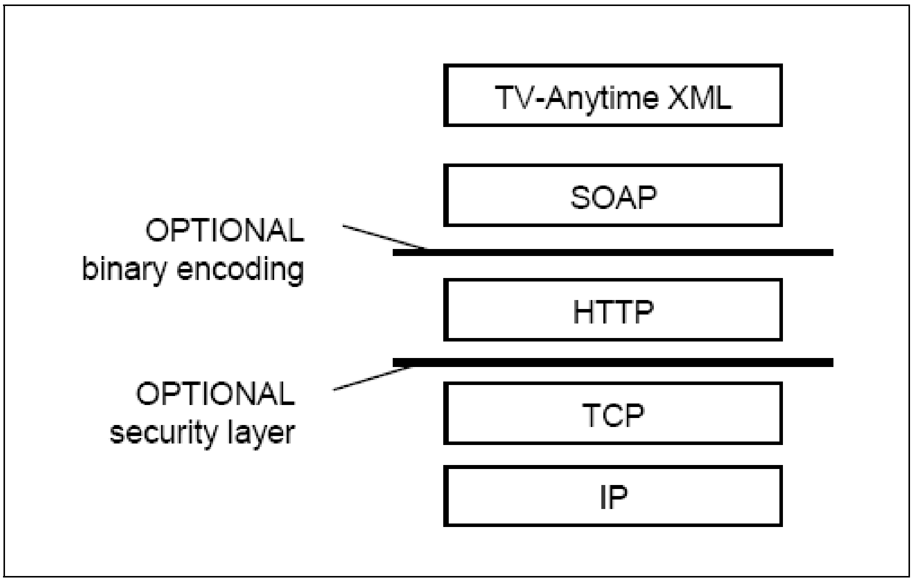

Figure 14: BCG over SOAP Protocol Stack

For further details on the specific usage of SOAP see TS 102 822-6-1 [5], clause 6.1 SOAP. HTTP v1.1 shall be supported as specified in TS 102 539 [3].

As can be seen in figure 14 both encoding and security are optional. For the case of encoding the standard HTTP encoding negotiation should be used (Accept-Encoding header). Servers should always support no encoding to ensure interoperability. If TLS is used for security, then this can be indicated via the HTTP@Location URL SD&S entry (the one with the HTTP@SOAP set to "true").

Although security may be used for retrieving metadata it is a particular issue for the submit data method, due to the need to ensure privacy of user specific data. In addition, a user should be involved in the decision as to whether to enable the submission of either anonymous or user specific data. This can be a trade-off between offering a personalized service versus user anonymity.

The protocols used require that polling be used by a client to check for metadata updates. Selection of the polling interval should be tuned to provide a balance between speed of update versus Server load.

10.2.2.3 Examples

The query mechanism contains a large degree of flexibility, thus enabling a BCG client to create either simple or complex queries in order to restrict the size of metadata response. The granularity required depends on the application and on resource limitations. Therefore a small number of possible request and response examples are provided.

In all examples the SOAP and HTTP wrappers are omitted for clarity.

The following is an example response to a describe_get_Data request. The request is not shown as it is simply an empty describe_get_Data SOAP method request. The response provides a description of the BCG provider along with its capabilities, such as domain, table types and specific fields supported.

```
<describe_get_Data_Result xmlns="urn:tva:transport:2005" xmlns:tva="urn:tva:metadata:2005"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" serviceVersion="1">
    <Name>Metadata Service</Name>
    <Description>A metadata service</Description>
    <AuthorityList>
        <Authority>domain1.co.uk</Authority>
    </AuthorityList>
    <AvailableTables xmlns:tvaf="urn:tva:transport:fieldIDs:2005">
        <Table xsi:type="ProgramInformationTable" canQuery="tvaf:CRID  tvaf:Synopsis tvaf:Title
tvaf:Keyword  tvaf:Genre"/>
        <Table xsi:type="ServiceInformationTable" canQuery=" tvaf:serviceID tvaf:Name
tvaf:ServiceURL"/>
        <Table xsi:type="ProgramLocationTable" canQuery="tvaf:CRID tvaf:serviceIDRef  tvaf:start
tvaf:end tvaf:PublishedStartTime tvaf:PublishedDuration">
            <AvailableLocations>
                <ServiceURL>dvb://1.1.1.1</ServiceURL>
            </AvailableLocations>
        </Table>
    </AvailableTables>
</describe_get_Data_Result>
```

Example 1 describe_Get_Data response

The following is an example query and response using the get_Data method. The query requests ProgramInformation for all programmes with a specific title ("Film1"). The test condition is omitted as equals is the default.

```
<QueryConstraints>
    <BinaryPredicate fieldID="tvaf:Title" fieldValue="Film1"/>
</QueryConstraints>
<RequestedTables>
<Table type="ProgramInformationTable"/>
</RequestedTables>
```

Example 2: Title Query

```
<tva:TVAMain xml:lang="eng" xmlns:tva="urn:tva:metadata:2005"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <tva:ProgramDescription>
        <tva:ProgramInformationTable>
            <tva:ProgramInformation programId="crid://123">
                <tva:BasicDescription>
                    <tva:Title>Film1</tva:Title>
                    <tva:Synopsis length="short">A film</tva:Synopsis>
                    <tva:Genre href="urn:tva:metadata:cs:ContentCS:2005:3.4.6">
                        <tva:Name>Action</tva:Name>
                    </tva:Genre>
                </tva:BasicDescription>
            </tva:ProgramInformation>
        </tva:ProgramInformationTable>
    </tva:ProgramDescription>
</tva:TVAMain>
```

Example 3: Title Query Response

The following is another example query and response using a get_Data request. The query requests all of the programmes broadcast by a particular service available over a 2 hour period. Although the Server performs the query across all possible data the required response data is restricted to only those tables requested, in this case the ProgramInformation and ProgramLocation tables. An exception to this is that a ServiceInformation table is always returned if there is reference to a service in the response e.g. in the ProgramLocation table.

```
<QueryConstraints>
<PredicateBag type="AND">
    <BinaryPredicate fieldID="tvaf:PublishedTime" fieldValue="2006-11-01T12:00:00Z"
test="greater_than_or_equals"/>
    <BinaryPredicate fieldID="tvaf:PublishedTime" fieldValue="2006-11-01T14:00:00Z"
test="less_than_or_equals"/>
    <BinaryPredicate fieldID="tvaf:ServiceURL" fieldValue="dvb://1.1.1.1"/>
    </PredicateBag>
</QueryConstraints>
<RequestedTables>
    <Table type="ProgramInformationTable"/>
    <Table type="ProgramLocationTable"/>
</RequestedTables>
```

Example 4: Query For Programmes Over A Two Hour Period

```
<tva:TVAMain xml:lang="eng" xmlns:tva="urn:tva:metadata:2005"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <tva:ProgramDescription>
        <tva:ProgramInformationTable>
            <tva:ProgramInformation programId="crid://124">
                <tva:BasicDescription>
                    <tva:Title>Title1</tva:Title>
                    <tva:Synopsis length="short">A film</tva:Synopsis>
                    <tva:Genre href="urn:tva:metadata:cs:ContentCS:2005:3.4.6">
                        <tva:Name>Action</tva:Name>
                    </tva:Genre>
                </tva:BasicDescription>
            </tva:ProgramInformation>
            <tva:ProgramInformation programId="crid://125">
                <tva:BasicDescription>
                    <tva:Title>Title2</tva:Title>
                    <tva:Synopsis length="short">A Soap Opera</tva:Synopsis>
                    <tva:Genre href="urn:tva:metadata:cs:ContentCS:2005:3.4.2.1">
```

```
                        <tva:Name>Soap opera</tva:Name>
                    </tva:Genre>
                </tva:BasicDescription>
            </tva:ProgramInformation>
        </tva:ProgramInformationTable>
        <tva:ProgramLocationTable>
            <tva:Schedule serviceIDRef="service1" start="2006-11-01T00:00:00Z" end="2006-11-
01T23:59:59Z">
                <tva:ScheduleEvent>
                    <tva:Program crid="crid://124"/>
                    <tva:PublishedStartTime>2006-11-01T12:00:00Z</tva:PublishedStartTime>
                    <tva:PublishedDuration>PT01H30M00S</tva:PublishedDuration>
                </tva:ScheduleEvent>
                <tva:ScheduleEvent>
                    <tva:Program crid="crid://125"/>
                    <tva:PublishedStartTime>2006-11-01T13:30:00Z</tva:PublishedStartTime>
                    <tva:PublishedDuration>PT00H30M00S</tva:PublishedDuration>
                </tva:ScheduleEvent>
            </tva:Schedule>
        </tva:ProgramLocationTable>
        <tva:ServiceInformationTable>
            <tva:ServiceInformation serviceId="service1">
                <tva:Name>service1</tva:Name>
                <tva:ServiceURL>dvb://1.1.1.1</tva:ServiceURL>
            </tva:ServiceInformation>
        </tva:ServiceInformationTable>
    </tva:ProgramDescription>
</tva:TVAMain>
```

Example 5: Query Response

Annex A (normative): Application Layer FEC Protection

A.1 Introduction

This annex to the DVB-IP Guidelines is intended to provide guidance for those intending to implement the optional DVB AL-FEC specification (Annex E of the DVB-IP phase 1 handbook [1]). Clause A.2 discusses issues around configuration of AL-FEC and A.3 summarizes the options for sending. Clause A.4 describes how layered multicast sending can be used to allow the amount of FEC overhead to be varied to suite the packet error rates experienced on individual connections for multicast delivery.

This annex includes two documents that were created by DVB as part of the evaluation process as clauses A.5, A.6 to A.9. Clause A.5 presents the evaluation criterion that were agreed before the selection process started and A.6 to A.9 are from the report on the evaluation process and give the rationale for the choice of the hybrid approach used in the DVB AL-FEC specification.

Some parts of this annex (mainly from clauses A.5 to A.9) have been included in a separate DVB bluebook on AL-FEC evaluations because a number of standards organizations and others requested sight of it before this version of the guidelines was published by ETSI.

The DVB AL-FEC code is defined only for the case of RTP transport. The defined UDP transport cannot support AL-FEC in a backwards compatible manner.

A.2 Configuring FEC protection

This clause provides a brief overview of the issues and parameters which must be considered when configuring FEC protection using the DVB AL-FEC code.

The two principle parameters that must be determined are as follows:

- the AL-FEC block size, or "protection period";
- the AL-FEC overhead.

The AL-FEC block size is the number of packets which are protected together as a block, or equivalently the time required to send those packets at the stream rate ("protection period"). The AL-FEC overhead is the amount of additional FEC data ("repair packets") that are sent as a fraction of the original data. For example, if the AL-FEC block size is 100 packets and 10 repair packets are sent for each block, then the AL-FEC overhead is 10 %.

In order to determine the AL-FEC protection required, a good understanding of the loss characteristics of the target network is required. In general, the objective of AL-FEC is to provide error-free reception over long periods of time (several hours) - loss events which cannot be corrected by the AL-FEC should therefore be very rare. This means that loss characteristics must be understood over long time periods.

The AL-FEC code operates on each FEC block independently. This means that loss characteristics must also be understood at a timescale equal to the FEC block size: averages over long time periods are not sufficient. For example, the average packet loss over a one hour period may be very low, but if many of the losses are concentrated in a short period of time they may still overwhelm the AL-FEC code.

The following clauses provide some general guidelines on the effect of configuration parameters targeted at correcting for burst and random loss respectively. In practice, losses are a combination of these two.

A.2.1 Correcting for burst losses

In order to correct for isolated burst losses, a number of repair packets greater than or equal to the worst expected burst loss must be provided for each FEC block. If only the AL-FEC base layer is used, then a number of repair packets equal to the worst expected burst loss must be provided. Note that for the base layer, the number of repair packets per block must be a divisor of the block size (in packets). This configuration will correct for isolated burst losses, but will often not correct randomly distributed losses and generally cannot correct for cases where multiple bursts occur within a block.

When the AL-FEC base and enhancement layers are used, it is generally sufficient to provide a number of repair packets one greater than the worst expected burst loss per block. This configuration generally can also correct for randomly distributed losses or multiple bursts per block provided sufficient enhancement layer packets are provided.

Note that the number of repair packets required per block is fixed independent of the block size. As a result longer block sizes will result in a lower relative AL-FEC overhead. However, longer block sizes will also contribute additional latency, affecting channel change times.

A.2.2 Correcting for random losses

In order to correct for randomly distributed losses, it is necessary to understand the "worst case" number of lost packets within an FEC block. If losses were truly random and independent, then the statistical probability of losing 1, 2, 3, etc. packets in a block could be calculated and from these probabilities the expected frequency of such events could also be calculated. The "worst case" of interest is then the worst case occurring frequently enough to be an issue from a quality perspective (which is a judgement issue on the part of the service provider). If this "worst case" can be corrected by AL-FEC, then although there may remain uncorrected events these will be so rare that they can be ignored (for example once a day, or once a week).

In practice, losses are not independent and random and so the worst case cannot be calculated statistically. Network measurements need to be used to determine the worst case that the AL-FEC must correct.

When only the base AL-FEC layer is used, then certain levels of random packet loss can be corrected. Annex B provides some simulated examples.

When base and enhancement layer AL-FEC is used, then the minimum number of repair packets needed to correct a worst case of *n* randomly distributed lost packets per block is *n*+1, where ope of the packets is a base layer packet and the remainder are enhancement layer packets. It should be noted that this configuration will not provide very much protection for end devices which support only the base layer. If more than one base layer packet is provided, for example to provide burst loss protection to devices which do not support the enhancement layer, then this reduces the effectiveness of the overall code in the face of random losses and more than *n*+1 repair packets may be needed in total.

Finally, although in this case the number of repair packets required is not independent of the block size, a similar trade-off exists between additional latency and bandwidth. For example, if the block size is 100 packets and the worst case loss is 10 packets, then it is highly unlikely that two 100 packet blocks, each with 10 lost packets should occur in sequence. In fact the worst case loss for 200 packet blocks may be only slightly larger than 10, meaning that the bandwidth overhead can still be roughly halved by increasing the block size from 100 packets to 200 packets. Note that because measurements are taken over very long time periods, and thus millions of blocks, then even if the average packet loss is very low, it may still occur that occasionally events such as 10 lost packets in a block of 100 occur.

A.3 FEC sending arrangement considerations

A.3.1 Introduction

Another important issue in the determination of FEC performance is the arrangement of data packets (source and FEC "repair" packets) in time for sending. The sending arrangement impacts FEC performance in three ways:

- The additional latency introduced by the use of FEC.
- The data rate profile (constant vs. bursty) of the resulting stream.

- The FEC overhead required to overcome packet loss with given characteristics.

The additional latency is impacted because it is necessary for the receiver to wait long enough for reception of all packets (source and repair) of the first source block before beginning presentation of the stream to the user. This is true even if there is no loss in the first block because once presentation has begun, and assuming freezing of the video is unacceptable, the presentation schedule for the whole stream is set by the initial start time. If presentation of the first block begins before all packets have arrived then presentation of every block will have to begin before all packets have arrived and this will prevent the FEC operation from being applied to recover losses when they do occur.

A sending arrangement which sends the FEC repair packets as soon as possible after the source packets minimizes this additional latency. Sending arrangements which interleave repair packets with source packets from the subsequent block increase the latency according to the amount of interleaving.

The sending arrangement clearly impacts the data rate profile. If FEC repair packets are sent in a burst immediately after each source block then the overall data rate profile will be very bursty.

Finally, the FEC overhead required may also be impacted. For example, when packets from a given block are sent in quick succession, then a burst outage may cause the loss of many packets. If they are spread out over time then fewer packets (from that block) will be lost. The FEC overhead is often dimensioned based on anticipated worst case burst outages and thus the sending arrangement can impact the required overhead.

A.3.2 Client considerations

The DVB-IP AL-FEC standard does not prescribe a particular sending arrangement: sending devices are free to use whatever sending arrangement they choose, subject to certain constraints. Receivers should be able to process incoming packets whatever arrangement they arrive in.

The service discovery signalling for FEC protected streams may provide information about the stream which may be used by receivers to determine the amount of buffering required for FEC purposes. The "FEC Maximum Block Size (Packets)" indicates the maximum number of stream source packets that will occur between the first packet of a source block and the last packet for that source block (source or repair). A receiver may keep a count of the number of source packets which have been received since the first packet of a particular source block. This count should include packets from any blocks, not just the particular one of interest. Once this count reached the signalled Maximum Block Size (Packets), the receiver may assume that no further packets (source or repair) for the particular source block will be received. It is then safe to begin presentation of the block. This approach is applicable in cases where the stream is constant bit-rate and the source blocks of constant duration.

The SD&S signalling may alternatively or additionally indicate the "FEC Maximum Block Size (Time)". This indicates the maximum sending duration of any FEC block. A receiver may measure the elapsed time from the receipt of the first packet of a particular block. Once this time exceeds the Maximum Block Size (Time) the receiver may assume that no further packets (source or repair) for the particular source block will be received. It is then safe to begin presentation of the block. This approach is applicable in cases where the stream is constant or variable bit-rate and the source blocks are of constant or variable duration.

Since IP networks introduce jitter, receivers should not make assumptions based on short-term measurements of packet arrival times. Long-term measurements can yield reliable information about clock drift between sender and receiver, but otherwise, clock recovery at receivers should be based on RTP timestamps and MPEG-2 Program Clock References, not on the absolute packet arrival times.

A.3.3 FEC Sending Arrangements

This clause describes some possible FEC Sending Arrangements. Other arrangements are possible. Receiver implementations should not make assumptions about the sending arrangement in used, except that it will conform to the signalled Maximum Block Size.

A.3.3.1 Constant rate, non-interleaved sending

In this sending arrangement, depicted in figure A.3.3.1, the overall sending rate is kept constant and the source packets of each block are sent before any of the repair packets of the block. This approach requires that the sending rate of the source packets be increased marginally to make space for the repair packets at the end of the block.

It is important to note that the sequencing of packets is determined by the FEC procedures which operate "below" the RTP layer. The contents of the packets, in particular the RTP timestamps, should not be modified compared to the case in which FEC is not applied so that the correct timing for the packets can be reconstructed with the usual procedures.

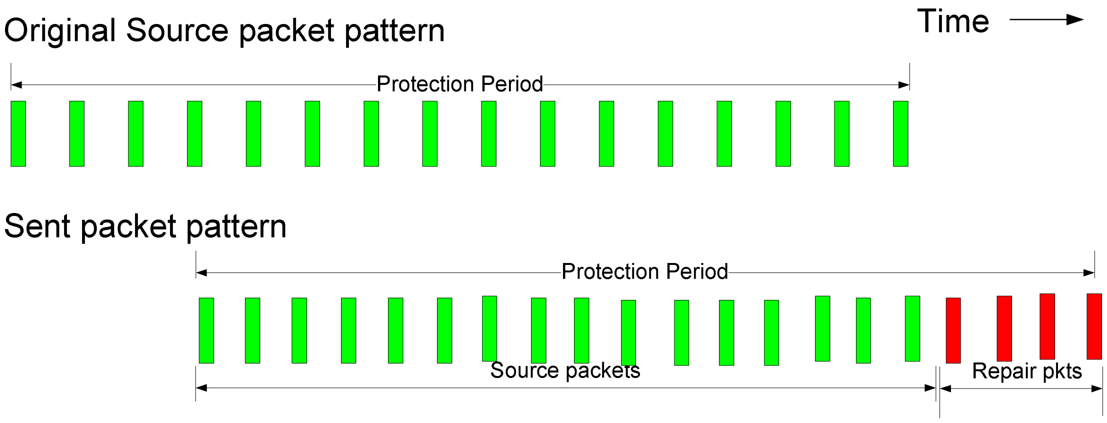

Figure A.3.3.1: Constant-rate, non-interleaved sending arrangement

Advantages of this sending arrangement are that the total data rate remains constant and the additional latency due to FEC is minimized. However, insertion of repair packets introduces small amount of jitter on all source packets.

A.3.3.2 Fully interleaved sending

In this sending arrangement, depicted in figure A.3.3.2, the overall sending rate is kept roughly constant and the sending rate of source packets is also kept constant.

Because this sending arrangement distributes repair packets for one block over the entire duration of the next block, then the additional latency due to FEC is equal to the duration of two blocks. When working with a fixed latency budget, this implies that the block size for the sending arrangement described here would be half that for the sending arrangement described in clause A.3.3.1. As a result, the overhead required by the code is increased.

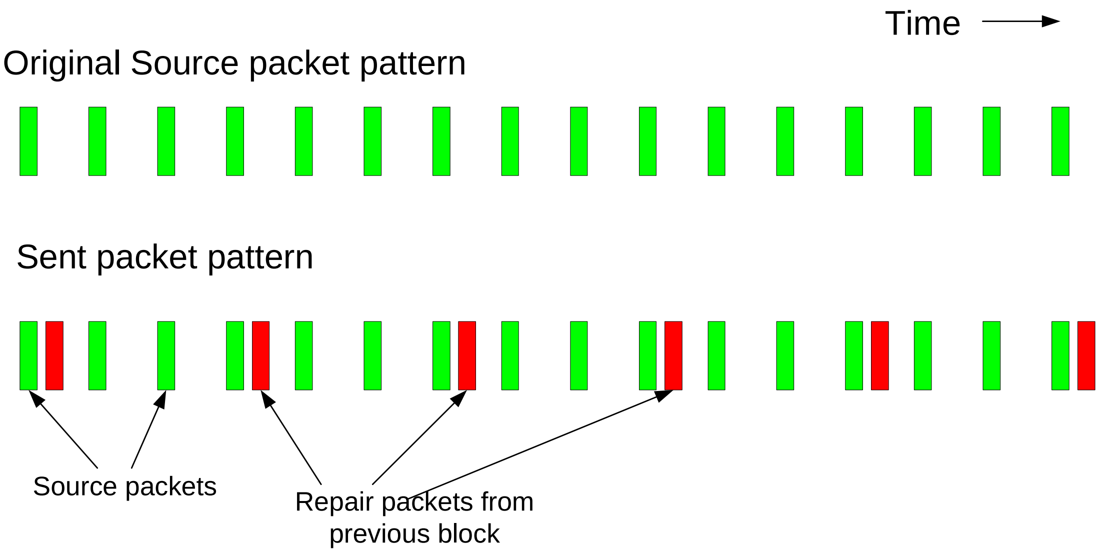
Time
Original Source packet pattern
Sent packet pattern
Source packets
Repair packets from previous block

Figure A.3.3.2: Fully interleaved sending arrangement

An advantage of this approach is that, except for small perturbations caused by the introduction of the repair packets, then the arrival times of the source packets are similar to the arrival times when FEC is not used. Additionally, the overall data rate is roughly constant. However, the high latency with respect to the block size is a significant issue.

A.3.3.3 Partially interleaved sending

In this sending arrangement, depicted in figures A.3.3.3, repair packets for one block are interleaved with the first few packets of the next block. As a result, the instantaneous sending rate during these first few packets is significantly increased. However, the block size may now be set almost as large as the latency budget, which reduces the required overhead.

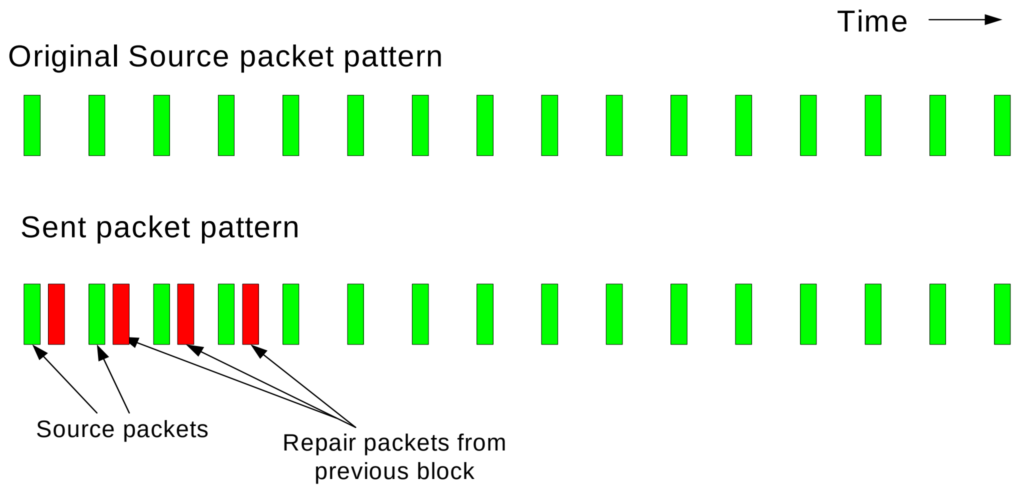
Time
Original Source packet pattern
Sent packet pattern
Source packets
Repair packets from previous block

Figure A.3.3.3: Partial interleaving sending arrangement

An advantage of this arrangement is that the source block size may be almost as large as the latency budget. However, the sending rate is extremely bursty - with double the bandwidth used at the beginning of each block. Note that if traffic shaping is used to return the stream to constant bit-rate, this will introduce jitter similar to that introduced by the constant, non-interleaved sending arrangement of clause A.3.3.1. However, the additional latency will still be higher than in the constant non-interleaved case.

A.3.3.4 Faststart sending for stored/buffered content

This sending arrangement, applicable to stored or buffered content (i.e. VoD and trick modes on live content) is illustrated in figure A3.3.4. In this arrangement, source data is sent slightly faster than the nominal stream rate at the start of the session or when trick modes are used. This allows the buffering period to be gradually increased without introducing additional latency.

Two variants of this approach are described here:

- "faststart with constant rate sending" - in this approach the additional source data bandwidth is obtained by reducing the FEC bandwidth at the beginning of the stream. As a result the total stream rate remains constant, but stream quality is reduced for these few initial seconds.

- "faststart with variable rate sending" - in this approach the overall stream rate at the beginning of the stream is somewhat higher than the nominal stream rate (e.g. 20 % higher) for the initial few seconds of the stream, but as a result the stream quality is maintained.

During the DVB FEC evaluation exercise, the second approach provided the best results.

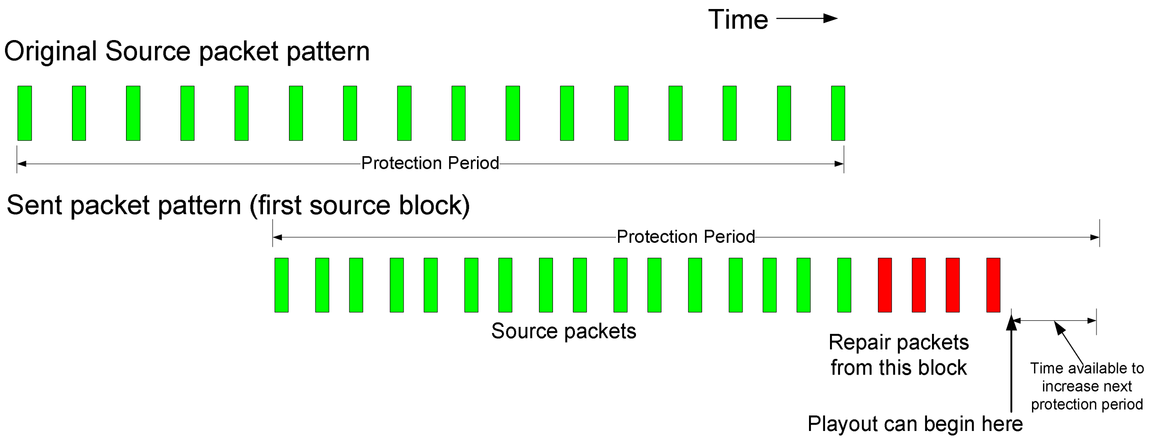
Time
Original Source packet pattern
Protection Period
Sent packet pattern (first source block)
Protection Period
Source packets
Repair packets from this block
Time available to increase next protection period
Playout can begin here

Figure A.3.3.4: Faststart sending arrangement

An advantage of this approach is that the source block size can quickly be increased without introducing additional latency. For example, the additional latency budget for FEC might be set at 100 ms, but by using the arrangement above the protection period can be increased to 500 ms over the first few seconds of the stream.

A.4 Layered multicast sending

The AL-FEC code defined by DVB supports layered sending in the multicast case. When layered multicast is used, then multiple multicast groups are used for a single DVB-IP stream. Each multicast group introduces incrementally more FEC protection so that receivers can adjust the amount and type of FEC data received according to their capabilities and requirements by joining and leaving the appropriate multicast group.

Source and repair packets within a DVB IP stream are "self identifying", meaning that the type and meaning of each packet can be identified from the packet contents (and in particular the UDP destination port number), without reference to the multicast group on which the packet was received.

DVB AL-FEC transmissions consist of a base layer and optionally one or more enhancement layers. Each layer may be provided on a different multicast group. The IP multicast group and destination UDP port number for each layer are provided within SD&S signalling.

Receivers which do not support or do not require FEC data should join only the multicast group associated with the original stream. Those receivers supporting or requiring only the FEC base layer should additionally join the multicast group associated with the base layer and those receiver supporting or requiring the enhancement layer or layers should join the multicast group or groups associated with the enhancement layer. Where multiple groups are advertised, receivers should join them "incrementally" - i.e. they should join multiple groups rather than choosing a single group.

Receivers may determine the amount of AL-FEC required based on measurements of packet loss. However, since the AL-FEC is designed to deliver a broadcast-quality stream the protection must be sufficient to handle even relatively rare packet loss events and so any such measurements must be over a long period of time. Alternatively, the number of layers to receive may be determined by operator configuration possibly linked to remote management.

The AL-FEC standard does not prescribe how much FEC overhead is allocated to each layer, nor the number of layers or the allocation of layers to multicast groups. In fact, all FEC data (base and enhancement layers) may be sent on the same multicast group as the original data or there may be one multicast group for original data and one for FEC data (base and enhancement layers).

A.5 Criterion for selection of Forward Error Correction for the protection of audiovisual streams delivered over IP Network Infrastructure

A.5.1 Requirements

Audiovisual services delivered over networks are subjected to the inherent properties of those networks including latency and errors. DVB commercial requirement is quoted as:

"Inclusion of suitable error protection strategies such as an FEC mechanism to enable DVB services to be carried over typical IP access networks with an acceptable quality of service (maximum 1 visible artefact/hour).

- *The selected solution shall be in line with work of other standards bodies such as DSL-Forum. If necessary, DVB should liaise with relevant other bodies.*
- *The selected solution shall provide flexibility so that it covers a reasonable range of networks and a variety of business models (trade-off versus payload). Furthermore, the selected solution shall be extensible to cover likely future streaming requirements.*
- *The selected solution shall be implementable on a range of HNEDs without significantly increasing product cost."*

The DVB TM agreed that the IP Infrastructure group should recommend an (optional) application layer FEC. It is agreed that it should work end to end including the core and home network where required

The FEC scheme selection process should take into account:

1) Packet loss characteristics of practical IP access network implementations e.g. DSL. These might include the use of interleaving at the physical layer to improve transport performance.

2) Further packet losses that could occur in the core network due to congestion and/or the home environment e.g. wireless technologies.

3) Sensitivity of A/V coding to errors.

4) Practical viability and flexibility of FEC scheme (encoding and decoding) to meet the min and max correction at minimal cost (processing, memory) for large numbers of simultaneous streams.

5) Ongoing cost of bandwidth inefficiency inherent in the code - i.e. difference between the bandwidth required by the code and the theoretical minimum bandwidth needed for service in the given loss conditions."

6) Pre-computation of the FEC to enable later usage when the content is streamed.

7) Carriage directly over RTP in the future i.e. without an MPEG2 transport stream.

8) Dynamically varying length of IP packets carrying A/V content.

A.5.2 System description

Figure A.5.2 is an example of video service delivery over DSL network from source (top left) to set top box (top right). It highlights the components through which the service is delivered and the logical position of the Application Layer FEC. Key points brought out by this diagram are:

a) There are other possible mechanisms that affect the delivery of acceptable quality of service (maximum 1 visible artefact/hour). These are DSL layer FEC/interleave, video/audio coding type and any error concealment at the decoder. The application layer FEC performance should provide adequate protection from errors **with and without** these mechanisms present (shown as min and max correction in figure A.5.2).

b) When these other mechanisms are present, the application layer FEC should take into account the effect of failure of these other mechanisms under severe error conditions.

c) When these other mechanisms are present, the 'load' on the application layer FEC is reduced under normal error conditions, leading to possible 'cost' reductions in terms of latency, memory, processor, etc.

d) Gaps in the core network domain and home network domain highlight the possible presence of other network types that could introduce service affecting packet loss. These networks should ideally be taken into account in the specification of application layer FEC performance, though will vary between implementations.

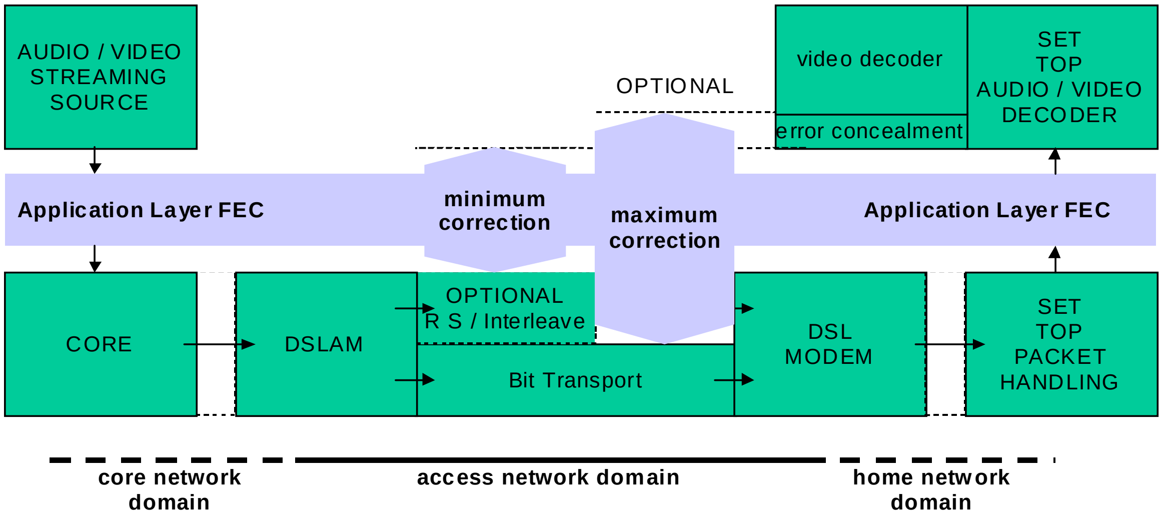

Figure A.5.2: minimum and maximum correction requirement for DSL access network domain

A.5.3 Packet loss characteristics

The packet loss characteristics should be provided by network operators and DSL chip vendors, ideally in the form of data collected from implementations or (if this is too commercially sensitive) in the form of a statement on what level of errors should be corrected by the application layer i.e. the requirements.

Worst case end-to-end packet loss metrics can be provided in terms of average loss rate, and loss distribution (independent random vs. bursty) for the IP packets, independent of bit rate. Note: methods for characterization of the loss distribution need further discussion.

Results for impulsive noise in DSL networks are available from the ITU and (until other information becomes available) they will be used as the basis of the evaluations. Although DSL is clearly an important case (where the results may vary widely), it is desirable to allow for other core, access and home networks also.

A.5.4 FEC Scheme Evaluation Criteria

Assume the following criteria:

1) Consider 3 error distributions: A. random losses (PLR 1e-3 to 1e-5), B. burst losses (PLR 1e-3 to 1e-5 with distributions based on ITU DSL results) C. better than 1e-5.

2) Additional latency due to FEC depending on applications (VOD = 100 ms, Broadcast = 400 ms).

3) Bit-rate for VOD = 2 Mbit/s, Broadcast = 2 Mbits and 6 Mbits (both based on H264/AVC.

4) Target mean time between FEC blocks that contain uncorrectable errors = 4 hours.

Data should be provided for each FEC proposal, specifying the performance for each set of parameters employed to illustrate range of performance available in terms of:

- Overhead required by the FEC to achieve the target performance in each of the given loss conditions (FEC data)/(protected data) (%).

- Flexibility:

  - Changing the overhead or/and the block size dynamically (within or between FEC blocks).
  - Range of protection periods.
  - Suitability for use with a wide variety of FEC sending strategies.
- STB memory requirement for buffering / processing (bytes).
- STB processing requirement measured as:
  - Maximum and average number of XOR operations.
  - Maximum and average number of conditional statements (IF..THEN).
  - Maximum and average number of context switching.
  - Maximum and average size of additional temporary memory needed.
  - Maximum and average number of threads (if threaded).
- Headend memory requirements for buffering (bytes).
- Headend processing requirement measured as:
  - Maximum and average number of XOR operations.
  - Maximum and average number of conditional statements (IF..THEN).
  - Maximum and average number of context switching.
  - Maximum and average size of additional temporary memory needed.
  - Maximum and average number of threads (if threaded).
  - Maximum memory bandwidth.
- Scalabilty, e.g. suitability for hardware implementation and cost.
- How much data is lost when the FEC fails? Visibility of artefacts when FEC fails.
- Ability to discard the FEC flow and process only the original packets as normal.
- Ability to add or remove FEC correction packets.

Additionally, systems considerations should be addressed including:

- Continued functioning of existing STB products in presence of FEC data.
- Option for new STB products to use or ignore FEC data.
- Confirmation of FEC scheme IPR compliance with DVB rules.
- Support of combined protection of audio and video packets.

A.6 AL-FEC evaluation report for DVB-TM IPI

This appendix contains the evaluation report of the DVB-TM IPI group on the proposed AL-FEC codes. Note that the two codes originally proposed were the Pro-MPEG Code of Practice 3 code as now specified in SMPTE 2022-1 [7] and the Digital Fountain Raptor code essentially as specified in TS 126 346 [6]. The eventually standardized code was a hybrid of these two original proposals.

A.6.1 Introduction

The report provides results of the DVB-TM IPI evaluation process for forward error correction for IPTV. Two candidate FEC codes have been considered, the Digital Fountain Raptor code, and the Pro-MPEG Code of Practice 3 based proposal.

Clause A.5 provides the agreed evaluation criteria, with the exception that it was later agreed to consider "additional latency due to FEC" of 100 ms and 400 ms (rather than "protection periods") and "mean time between packet loss" (rather than "mean time between FEC blocks with errors").

During the evaluation process, it was realized that a key issue in determining the FEC performance is the sequencing and timing of the sending of source and FEC packets. This issue is discussed further in clause A.6.2. Examples of sending arrangements are described in clause A.3.

This clause also includes simulation results for the following cases:

- "concurrent interleaved sending" - in which FEC packets are interleaved with the source packets they protect - these results are included in clause A.8.
- "hybrid code" - in which a mixture of Pro-MPEG and Raptor packets are sent - these results are included in clause A.9.

A.6.2 Sending arrangement considerations

An important issue in the evaluations was the way the different codes arrange data packets (source and FEC "repair" packets) for sending. Many different arrangements are possible for both codes. Since the arrangement can slightly impact the latency introduced by the FEC code with particular settings, and since these evaluations considered fixed latency budgets, the choice of sending arrangement affects the choice of parameters which are possible within the latency budget and therefore affects the bandwidth requirements of the codes.

An additional consideration with respect to sending arrangements is whether the resulting data stream has a constant bit-rate.

A.6.3 Bandwidth costs

A primary objective of the simulations performed as part of this evaluation exercise was to measure the bandwidth overhead required to achieve a target quality of service. Although not the only evaluation criteria for AL-FEC, bandwidth consumption represents an ongoing cost of the solution for the operator: excessive bandwidth consumption may translate into lower service quality, fewer services or a smaller target market.

In order to assess bandwidth requirements, simulations were performed according to the agreed cases. For each case, the simulated time was 96 hours and the *mean time between packet loss* was measured. The minimum bandwidth required was assessed by performing repeated simulations, gradually increasing the FEC overhead until the target mean time between packet loss was achieved. Note that in the case of the Pro-MPEG code, increasing the bandwidth required that a different code was used - i.e. change in the L and D parameters and possibly change in the type of parity packets sent: row, column or both.

A.6.3.1 Loss models

Two loss models were used in the simulations, independent random packet loss and a loss model based on DSL Repetitive Electrical Impulse Noise (REIN).

The REIN model results in fixed length (8 ms) burst losses which are randomly placed in order to achieve an overall loss rate within the 10^{-6} to 10^{-3} loss range of interest. As such, the results below for the REIN case give a good indication of the code performance in the presence of burst losses.

A.6.3.2 Multicast case

For the multicast case, a maximum additional latency of 400 ms was used. The graphs below show the FEC overhead required to achieve a mean time between packet loss of four hours, plotted against packet loss for both independent random packet loss and Repetitive Electrical Impulse Noise simulated. The overhead calculation is based on the actual number of bytes sent, including IP and other headers, not just the ration of repair packets to source packets.

The figures also include a plot for an "Ideal Block Code" - this represents the theoretical lowest overhead which could achieve the target quality within the maximum latency using a block FEC code and gives a useful guide as to how much of the bandwidth dedicated to FEC is actually needed to provide the required FEC protection and how much is overhead due to inefficiency in the FEC code itself.

Note that the overhead scale in each graph may be different, to show the range of interest.

A.6.3.2.1 Results with constant sending arrangement

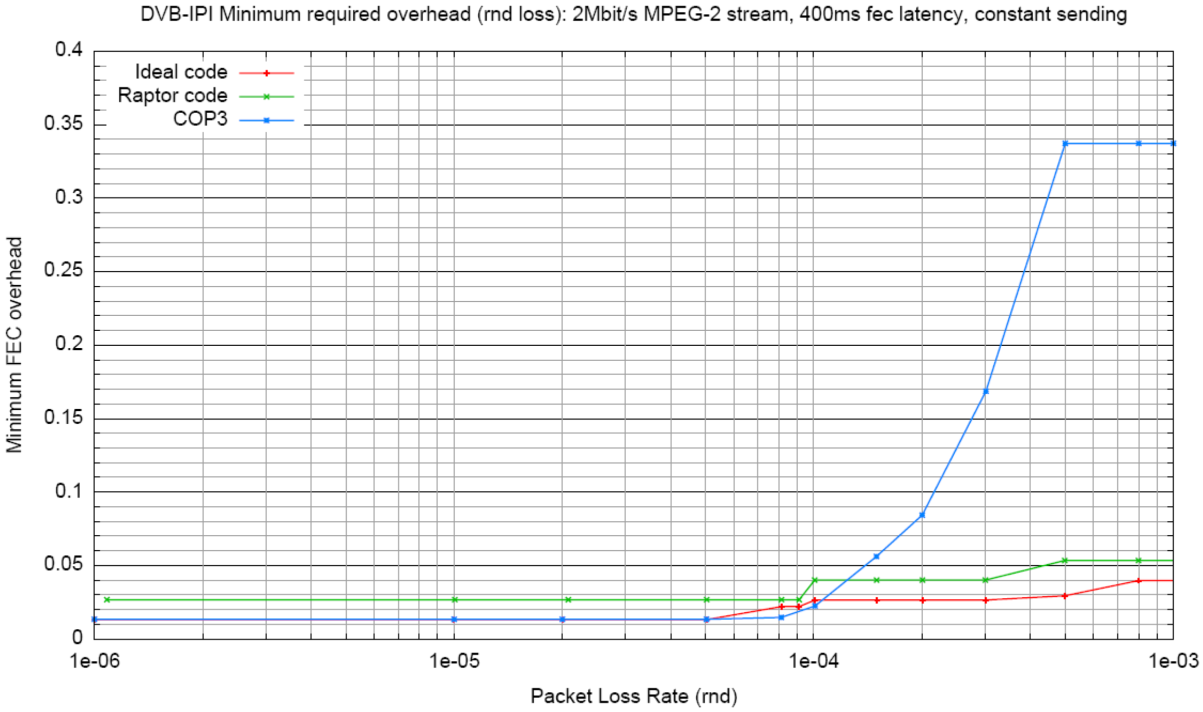

Figure A.6.3.2.1-1: 2 Mbit/s MPEG-2 Transport Stream, 400 ms latency, Random Loss, constant sending

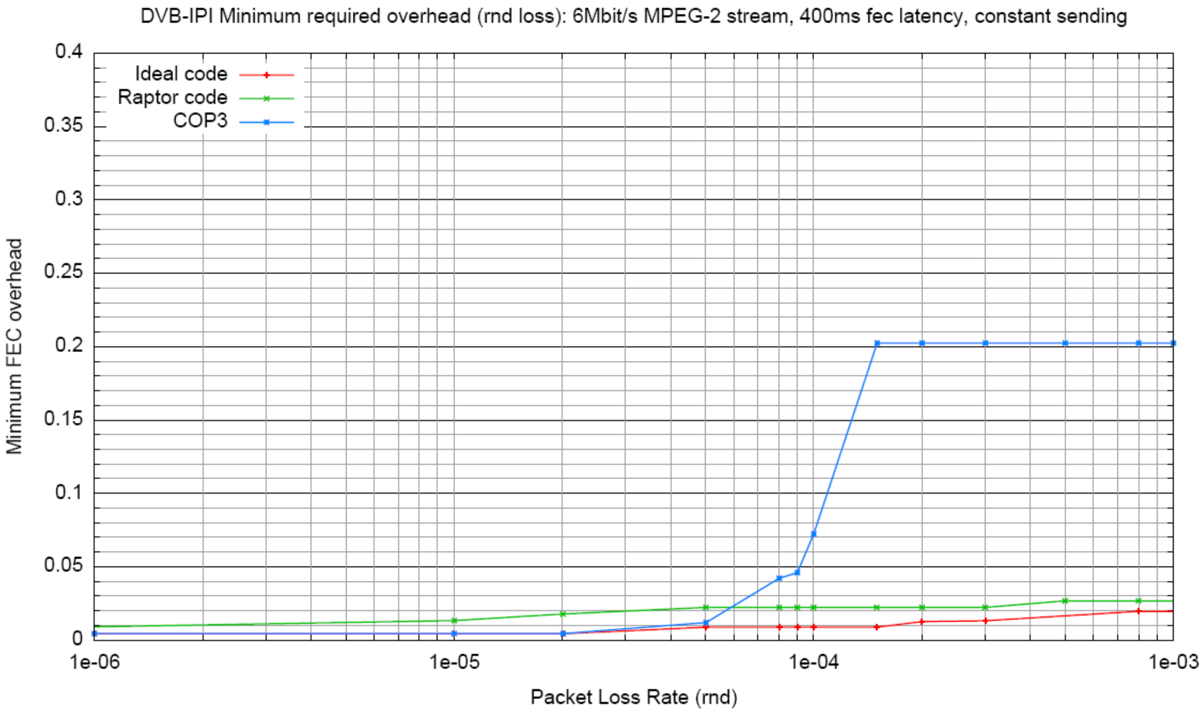

Figure A.6.3.2.1-2: 6 Mbit/s MPEG-2 Transport Stream, 400 ms latency, Random Loss, constant sending

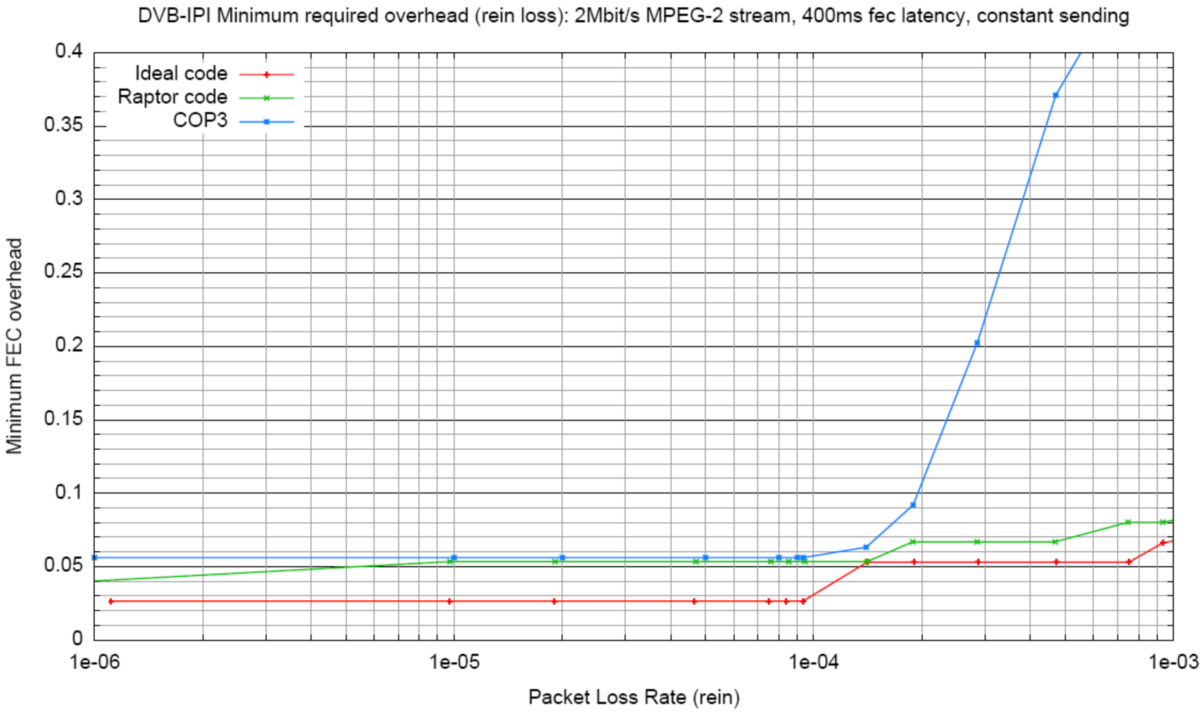

Figure A.6.3.2.1-3: 2 Mbit/s MPEG-2 Transport Stream, 400 ms latency, REIN, constant sending

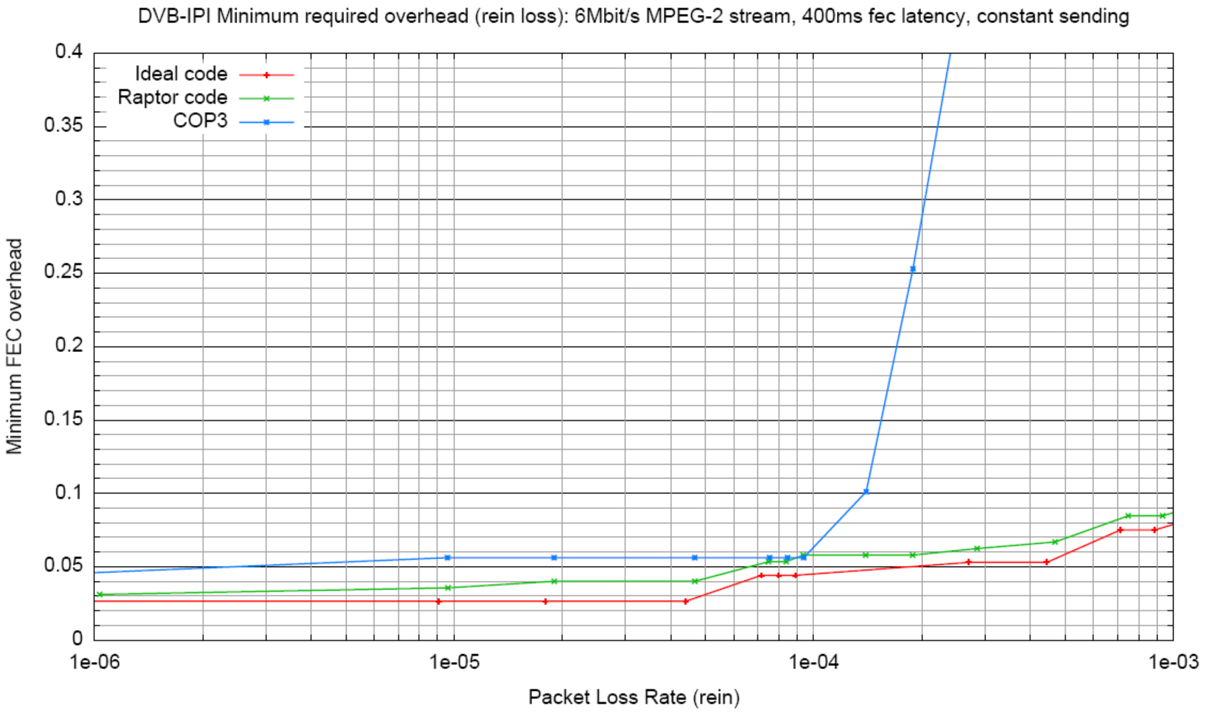

Figure A.6.3.2.1-4: 6 Mbit/s MPEG-2 Transport Stream, 400 ms latency, REIN, constant sending

A.6.3.2.2 Results with burst sending arrangement

NOTE: Curves for the "Ideal" block code and Raptor below are for constant rate sending, compared with burst sending for Pro-MPEG.

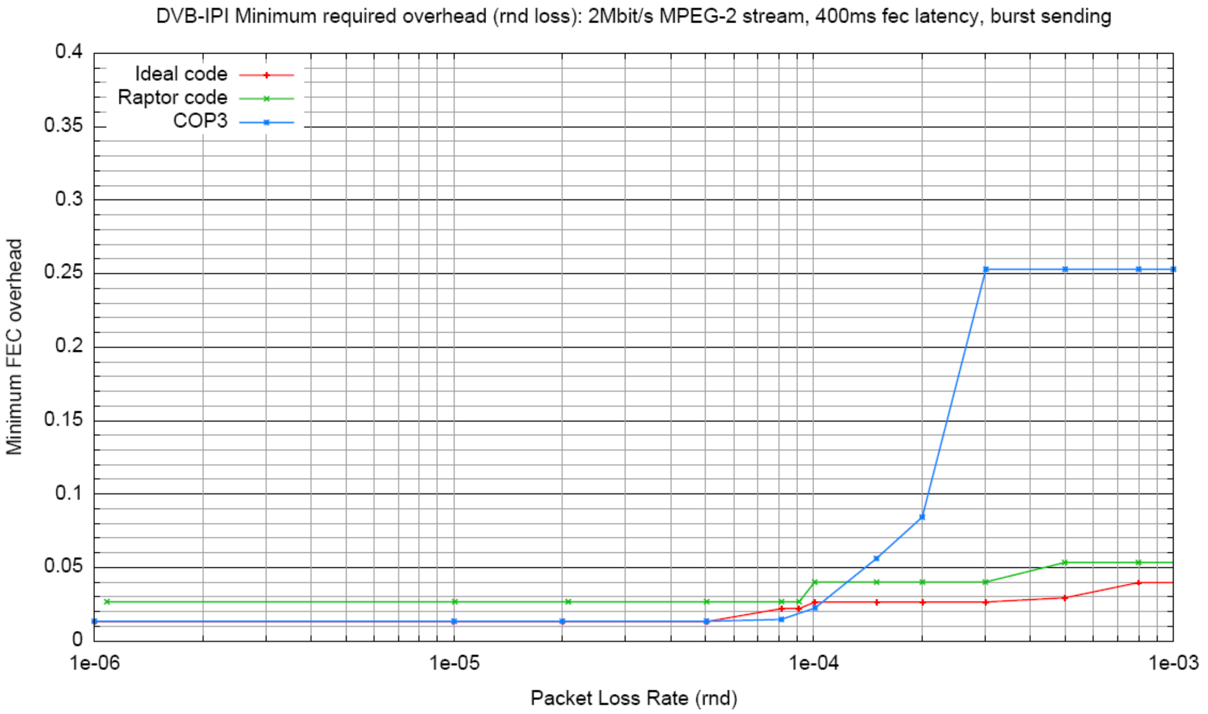

Figure A.6.3.2.2-1: 2 Mbit/s MPEG-2 Transport Stream, 400 ms latency, random loss, burst sending

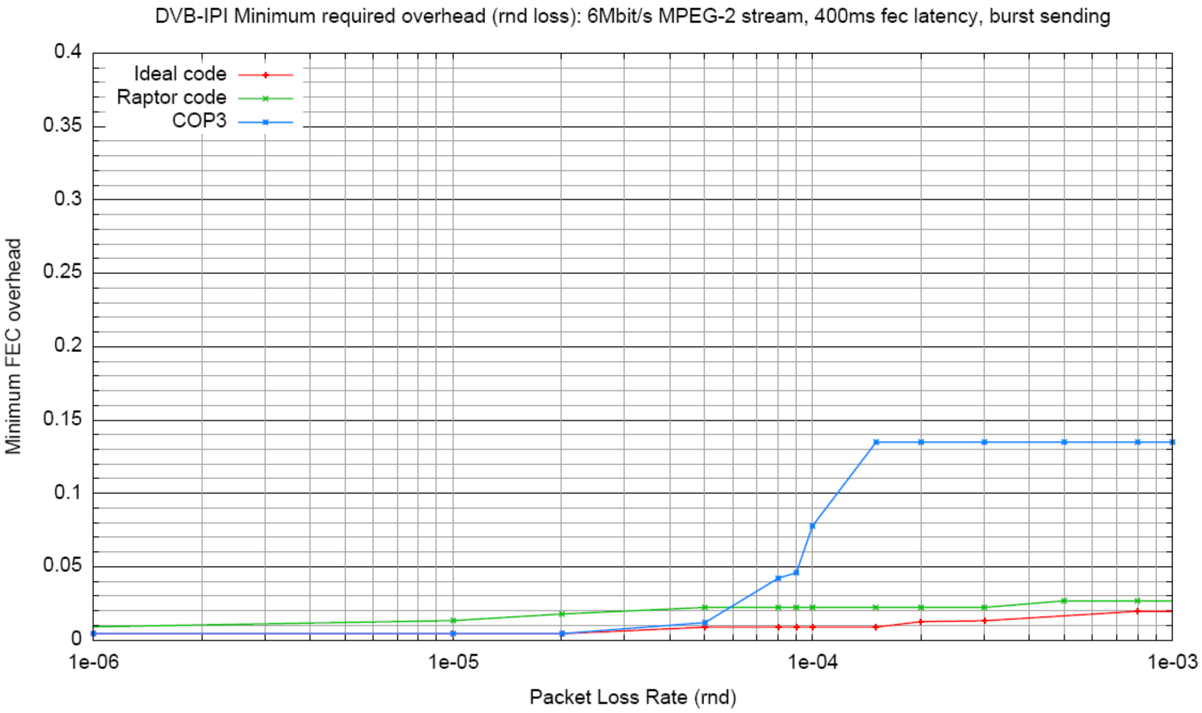

Figure A.6.3.2.2-2: 6 Mbit/s MPEG-2 Transport Stream, 400 ms latency, random loss, burst sending

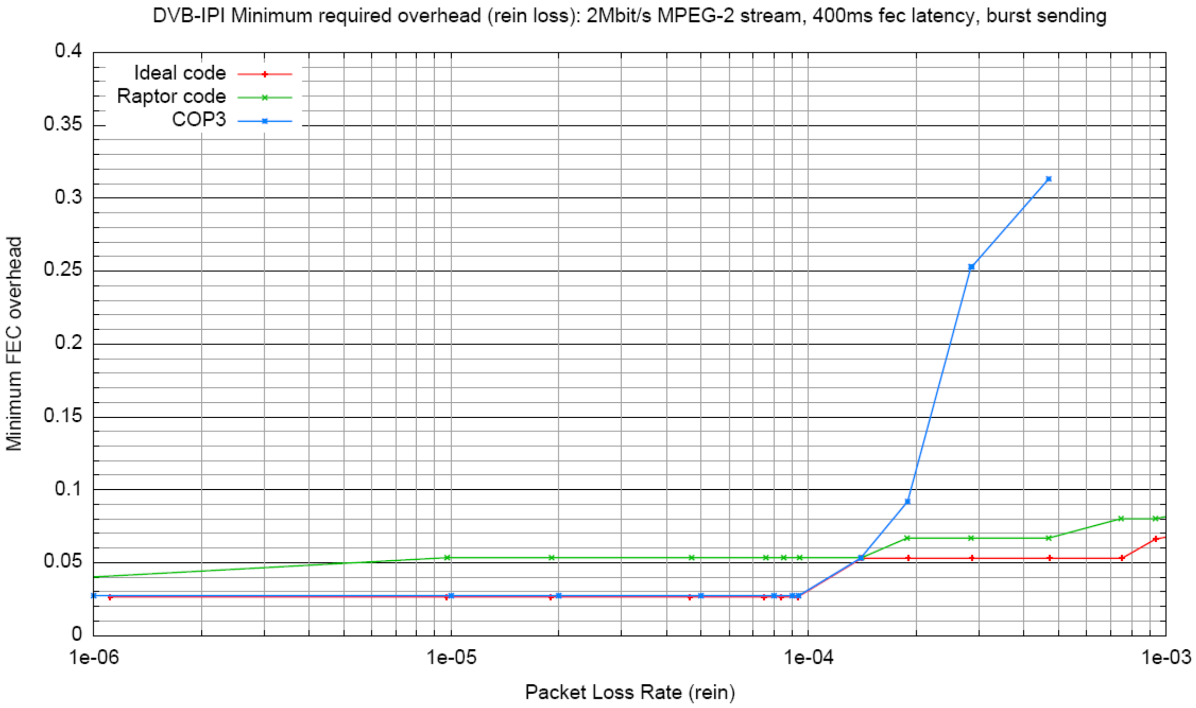

Figure A.6.3.2.2-3: 2 Mbit/s MPEG-2 Transport Stream, 400 ms latency, REIN loss, burst sending

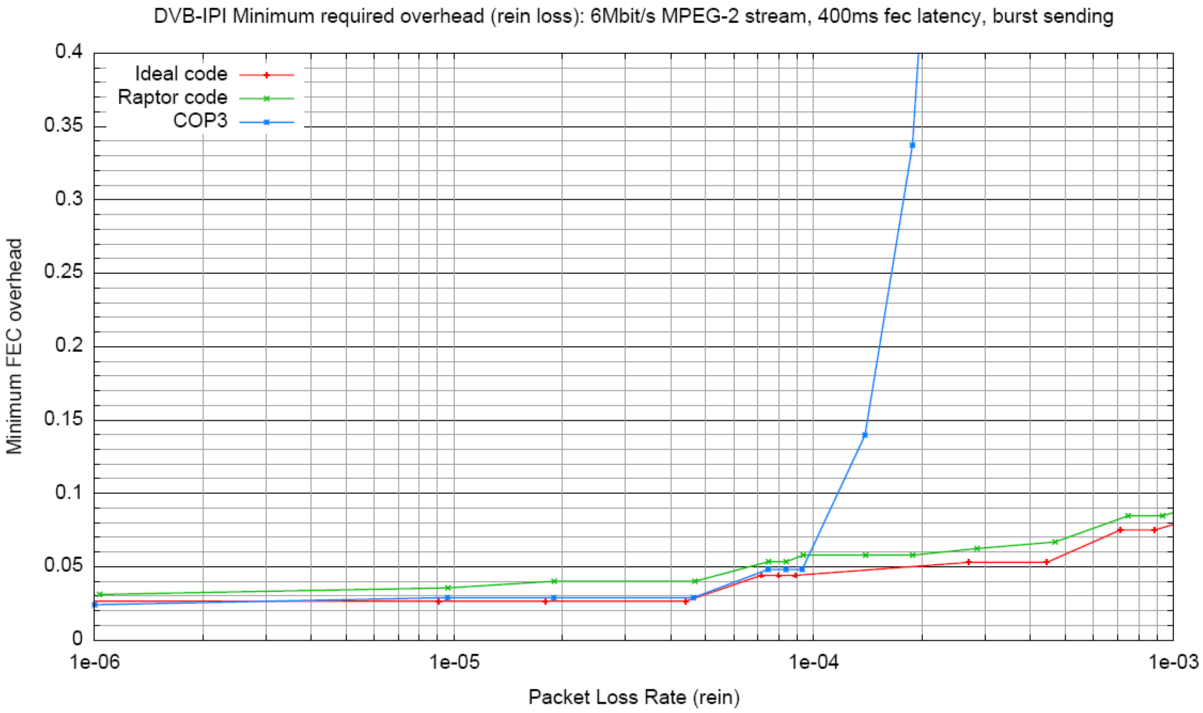

Figure A.6.3.2.2-4: 6 Mbit/s MPEG-2 Transport Stream, 400 ms latency, REIN loss, burst sending

We note the following from these simulation results:

- The Raptor code consistently requires close to the minimum possible overhead for a block code (as illustrated by the red "ideal" plots).

- The overhead required for the Raptor code increases smoothly as the loss rate increases.

- A modest Raptor overhead of 9 % provides for FEC protection up to above 10^{-3} packet loss in both the random and REIN loss models.

- The Pro-MPEG COP3 code with constant sending rate performs close to the ideal code whenever PLR remains under a threshold value around 10^{-4}.and only in the case of random loss this is the case since the Pro-MPEG row code is a simple parity code, which is optimal when only one packet of protection data is needed per block).

- Around 10^{-4} packet loss rate for the random loss case, the Pro MPEG code requires higher overhead - around 34 % for the 2 Mbit/s stream and 20 % for the 6 Mbit/s stream.

- Depending on the sending arrangement, above around 3 × 10^{-4} packet loss for the REIN case, no settings for the Pro-MPEG code which supported the required quality target (measured in mean time between packet losses) could be found. Nevertheless, when using a slightly lowest quality target (same time but measured in mean time between FEC blocks with errors), it is possible to find Pro-MPEG settings to support the required quality target.

- The burst arrangement for the Pro-MPEG code requires somewhat less overhead at high loss rates, although still significantly more than Raptor.

- The burst sending arrangement for the Pro-MPEG code offers significant improvements in the REIN case - in fact improving on the ideal block code (which uses a constant sending arrangement).

- The choice of burst or constant sending arrangement for Raptor makes little difference in the required overhead.

- The burst sending arrangement for Pro-MPEG does not allow the quality target to be achieved in the REIN case across the whole loss range. It should be noted that simulations based on a lower quality target *can* be met by ProMPEG.

It should be noted that in the above cases the parameters for the Pro-MPEG code were selected to provide the best performance for each particular loss rate and pattern through a wide search of the possible parameter set. In practice, we expect loss rates and error patterns to be largely unknown in advance.

In particular, for the REIN cases, the Pro-MPEG column code with a number of columns equal to the burst length provides adequate protection so long as events with two error bursts within a protection period happen only once every four hours or less.

This may happen when the overall loss rate is high or when there is strong correlation between bursts. Moreover if random single loss errors happen very close to a burst, they may not be corrected neither.

A.6.3.3 Unicast case

A.6.3.3.1 Stored/buffered content

In these cases, content is available at the server in advance of sending to the user: for VOD services the content is stored in its entirety and for live broadcast in trick modes the content is buffered for at least a few hundred ms when the user activates the trick mode by pausing the multicast broadcast.

In these cases the Raptor code incorporates a fast buffer fill technique (called "faststart" in this paper) which allows the protected block size to be gradually increased over the first few seconds of transmission. Note that this technique is possible only because of the independence of block size and overhead supported by Raptor and the possibility to flexibly vary the overhead in single packet increments without impacting the error correction performance of the code.

As above, repeated 96 hour simulations were performed with the FEC overhead again increased for each simulation until the target quality was achieved. The fast-start procedure is repeated every 10 minutes during the simulation to model the impact of repeated channel change or use of trick-modes.

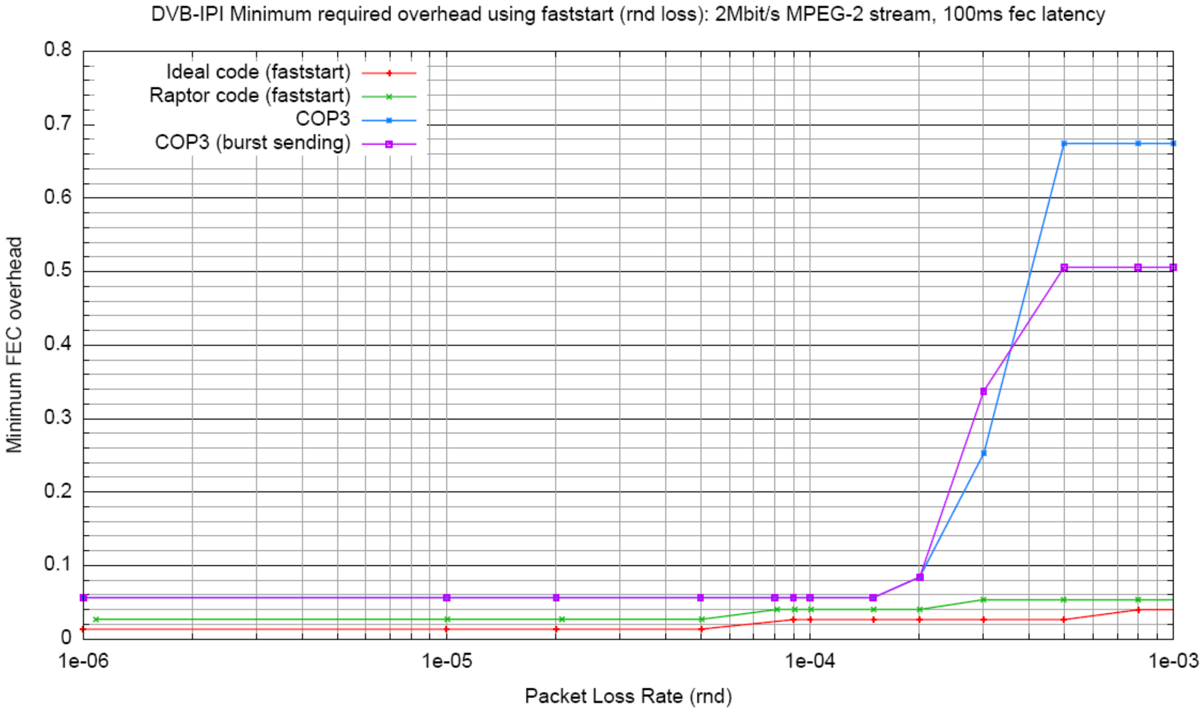

Figure A.6.3.3.1-1: 2 Mbit/s MPEG-2 Transport Stream, 100 ms latency (stored/buffered content), random loss

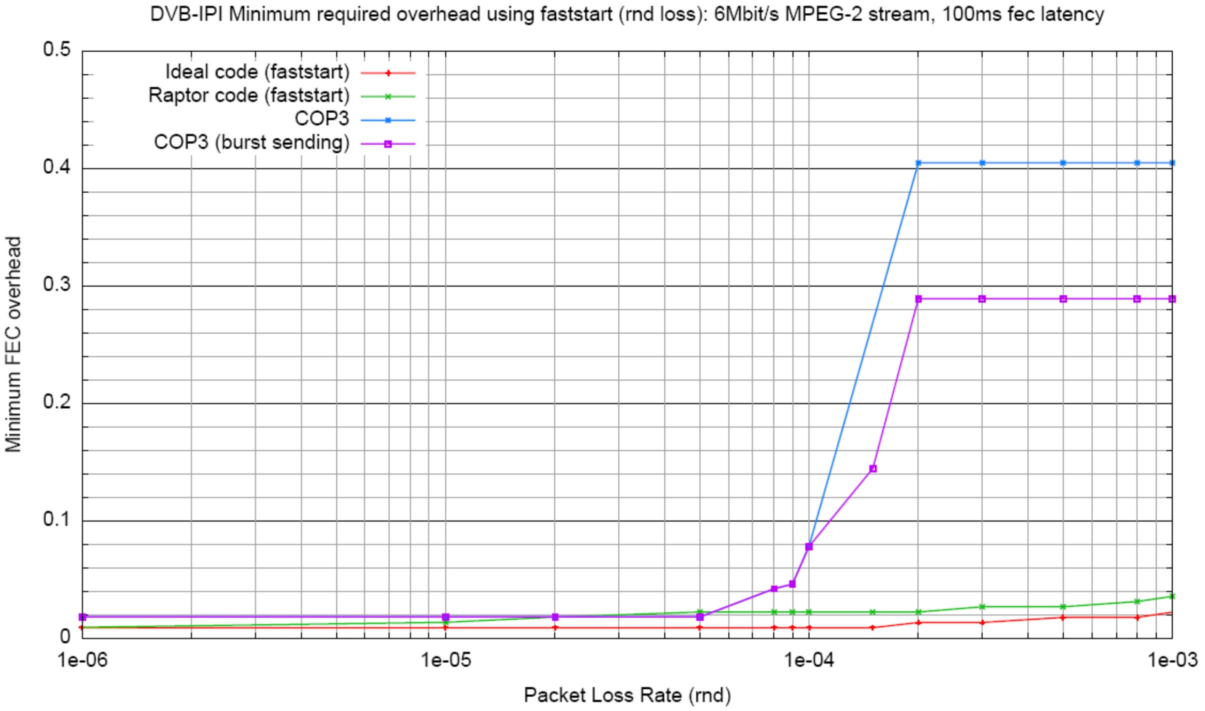

Figure A.6.3.3.1-2: 6 Mbit/s MPEG-2 Transport Stream, 100 ms latency (stored/buffered content), random loss

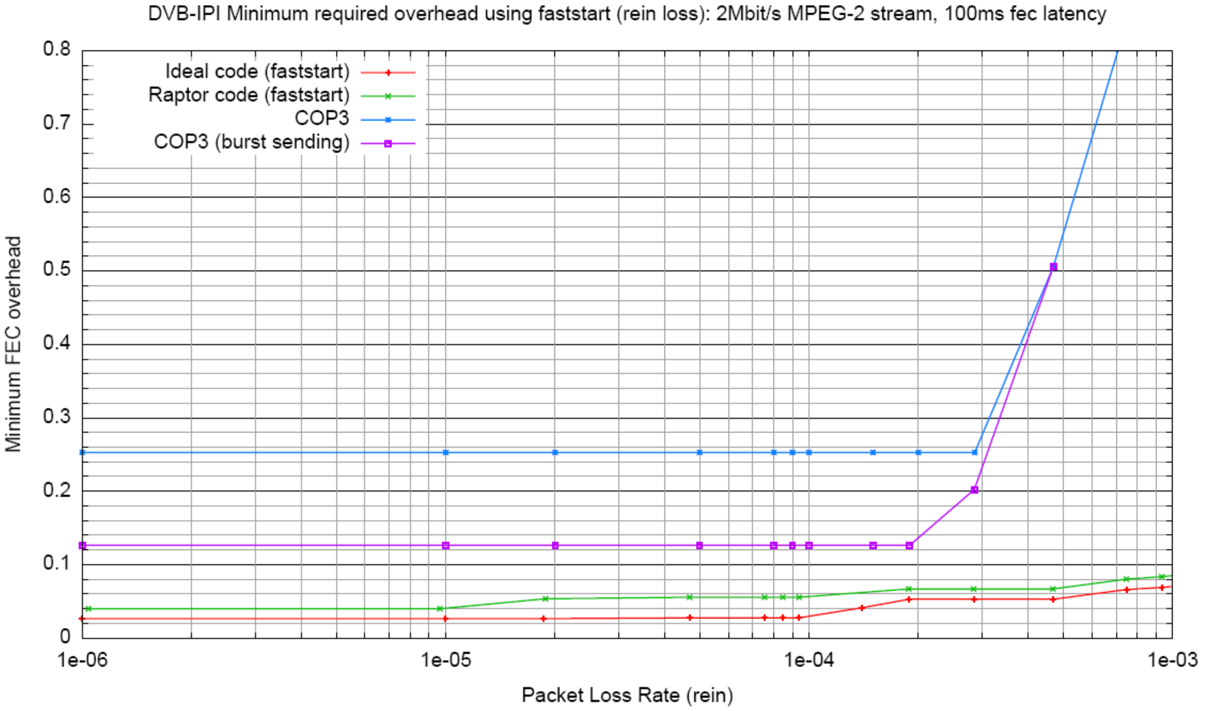

Figure A.6.3.3.1-3: 2 Mbit/s MPEG-2 Transport Stream, 100 ms latency (stored/buffered content), REIN

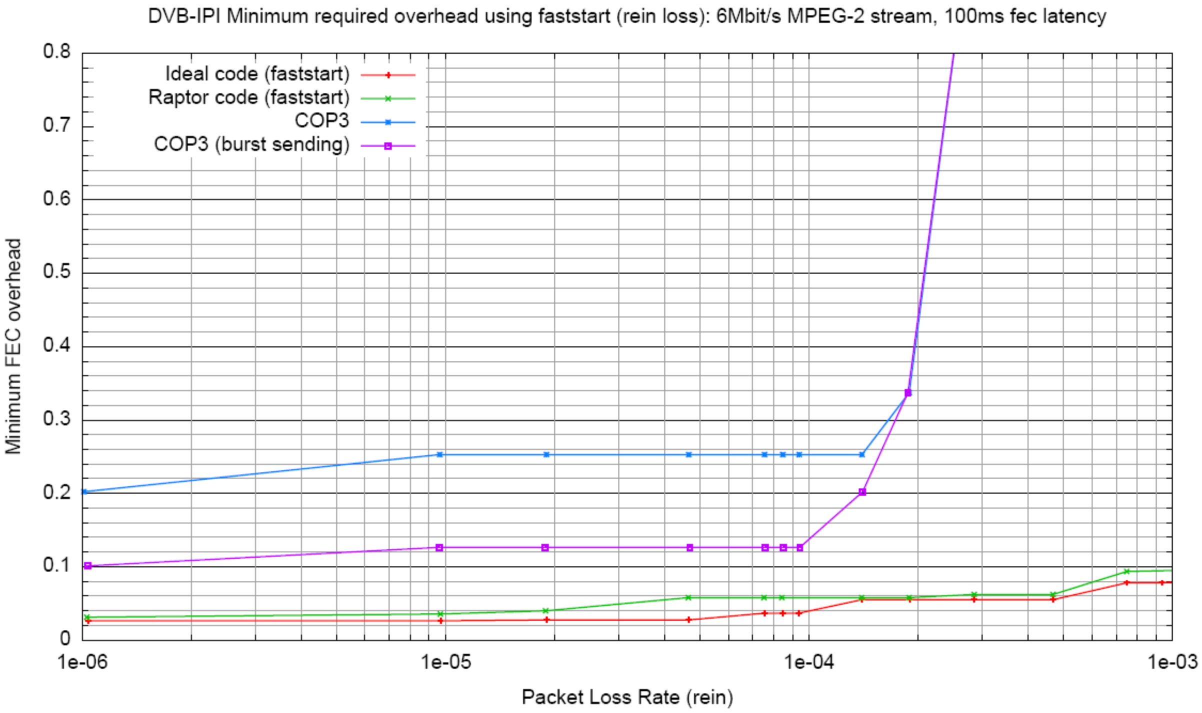

Figure A.6.3.3.1-4: 6 Mbit/s MPEG-2 Transport Stream, 100 ms latency (stored/buffered content), REIN

A.6.3.3.2 Live content

In the case of unicast delivery of live content (for example in networks which do not support multicast) then the block size for the Raptor code is limited by the requirement of a maximum latency due to FEC of 100 ms. The following figures show simulation results for this case.

A.6.3.3.2.1 Constant sending arrangement

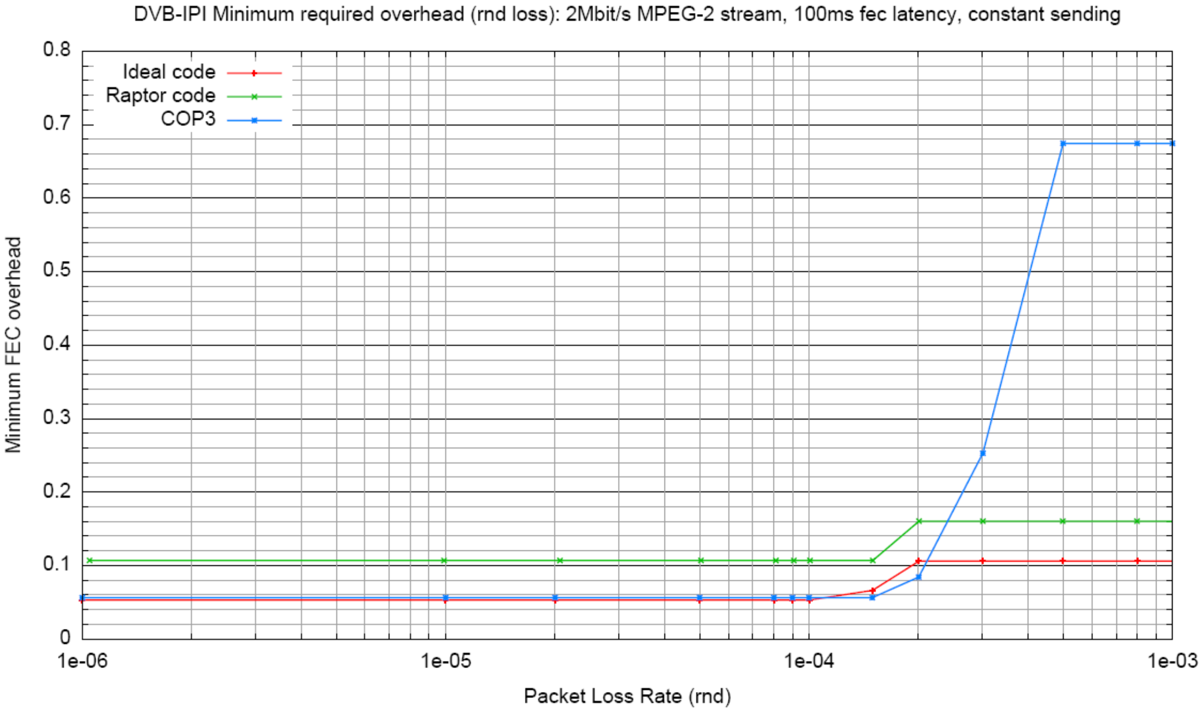

Figure A.6.3.3.2.1-1: 2 Mbit/s MPEG-2 Transport Stream, 100 ms latency (live content), random loss, constant sending

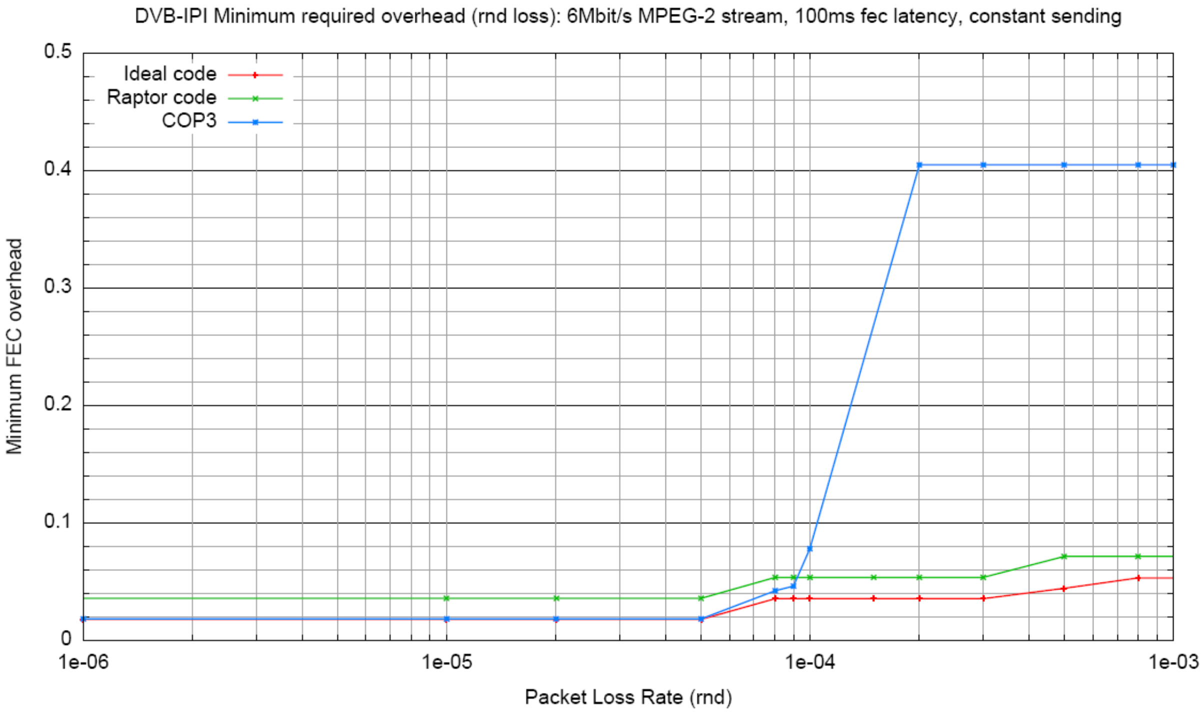

Figure A.6.3.3.2.1-2: 6 Mbit/s MPEG-2 Transport Stream, 100 ms latency (live content), random loss, constant sending

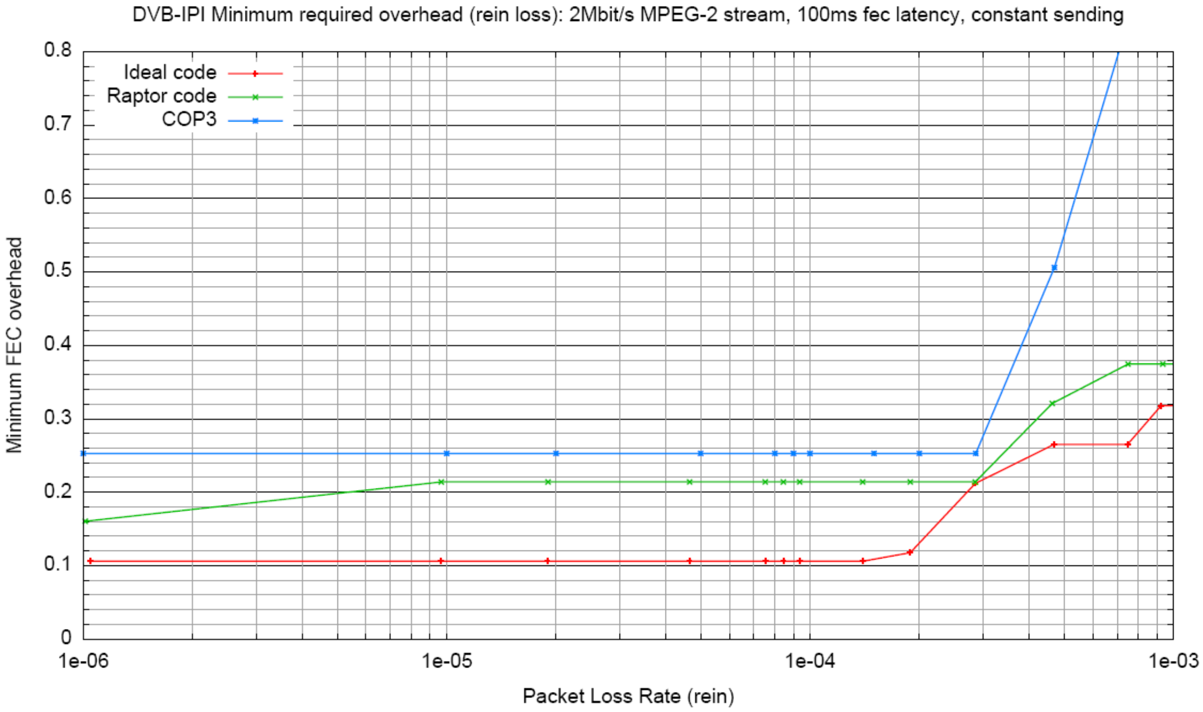

Figure A.6.3.3.2.1-3: 2 Mbit/s MPEG-2 Transport Stream, 100 ms latency (live content), REIN, constant sending

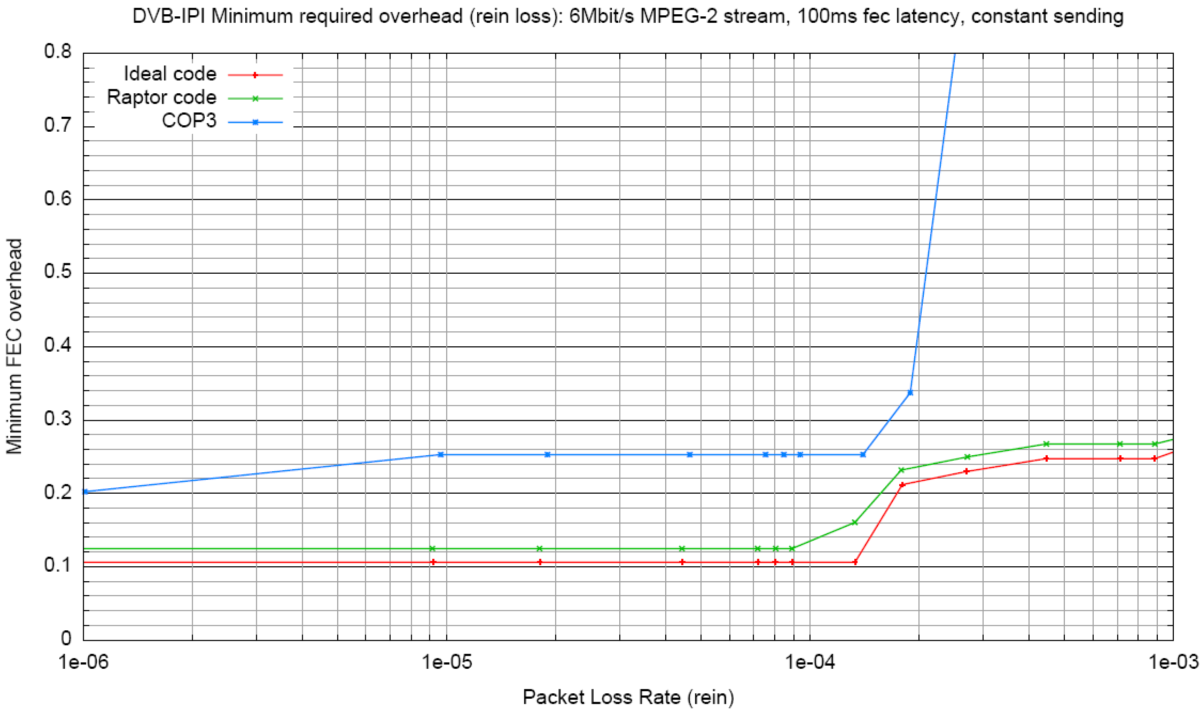

Figure A.6.3.3.2.1-4: 6 Mbit/s MPEG-2 Transport Stream, 100 ms latency (live content), REIN, constant sending

A.6.3.3.2.2 Burst sending

NOTE: Curves for the "Ideal" block code and Raptor below are for constant rate sending, compared with burst sending for Pro-MPEG.

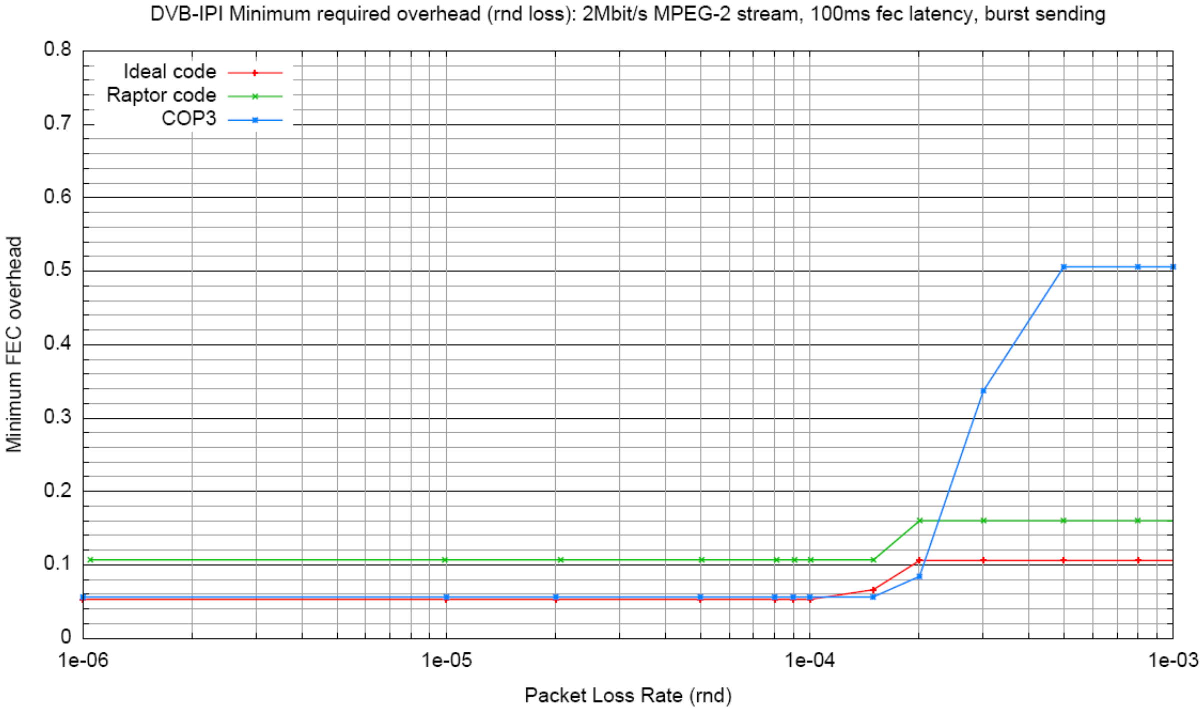

Figure A.6.3.3.2.2-1: 2 Mbit/s MPEG-2 Transport Stream, 100 ms latency (live content), random loss, burst sending

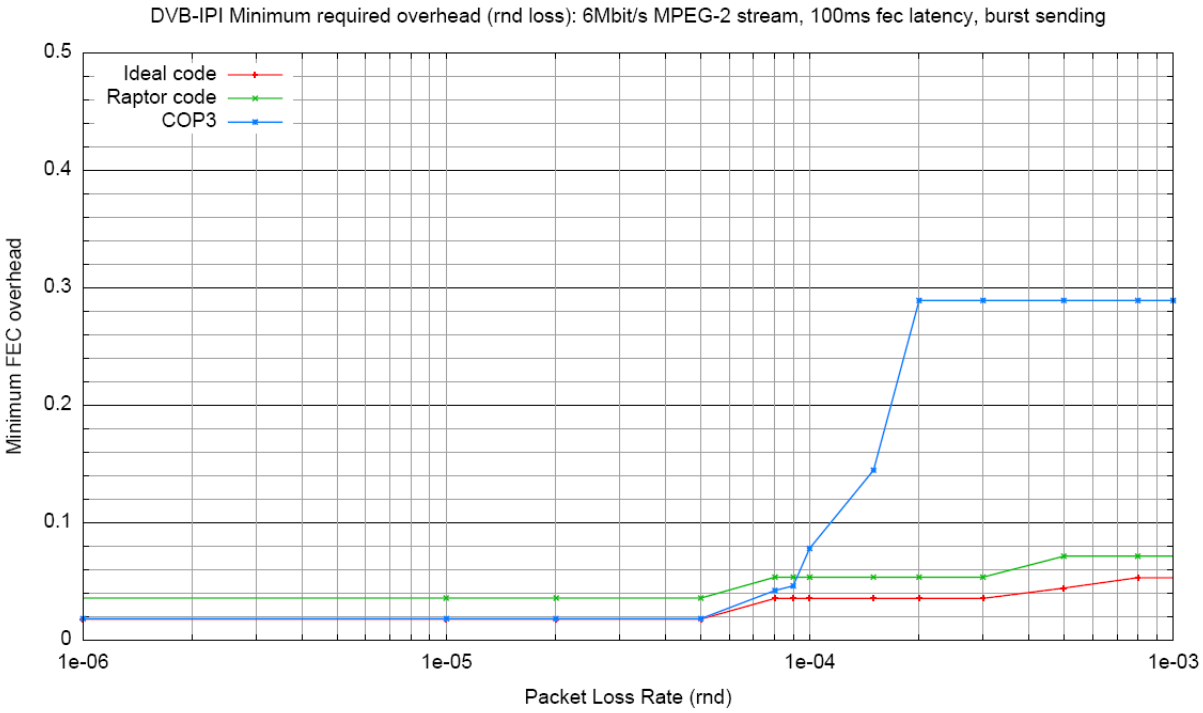

Figure A.6.3.3.2.2-2: 6 Mbit/s MPEG-2 Transport Stream, 100 ms latency (live content), random loss, burst sending

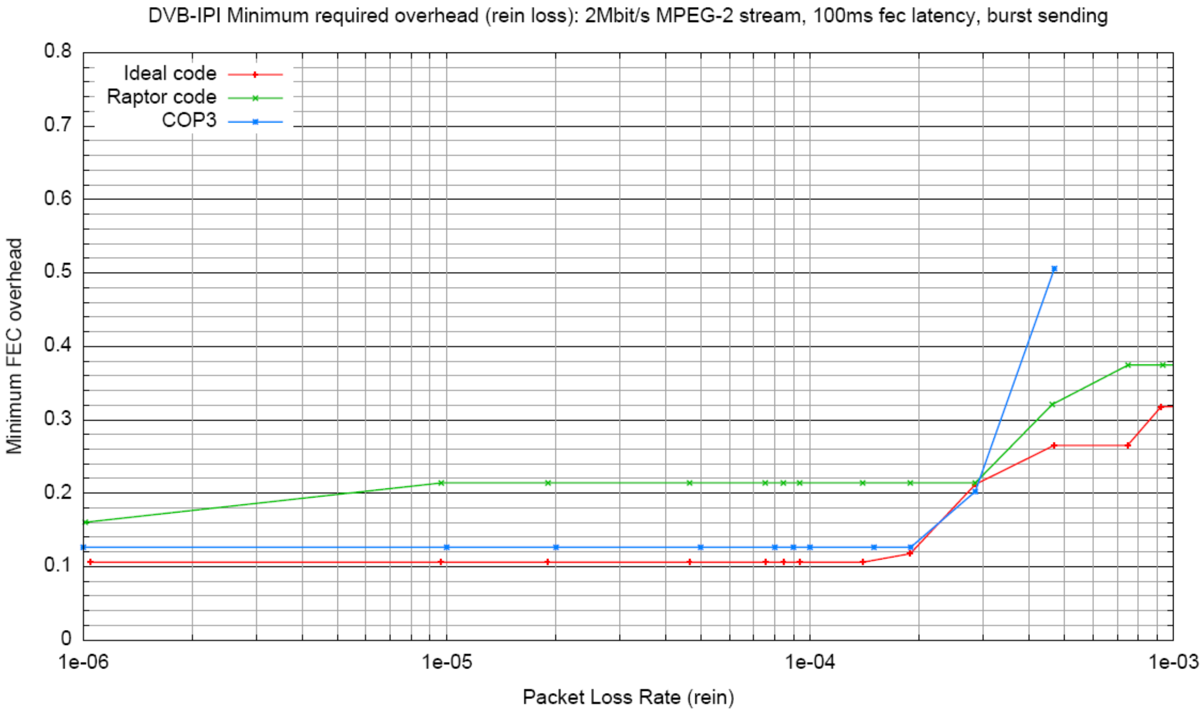

Figure A.6.3.3.2.2-3: 2 Mbit/s MPEG-2 Transport Stream, 100 ms latency (live content), REIN loss, burst sending

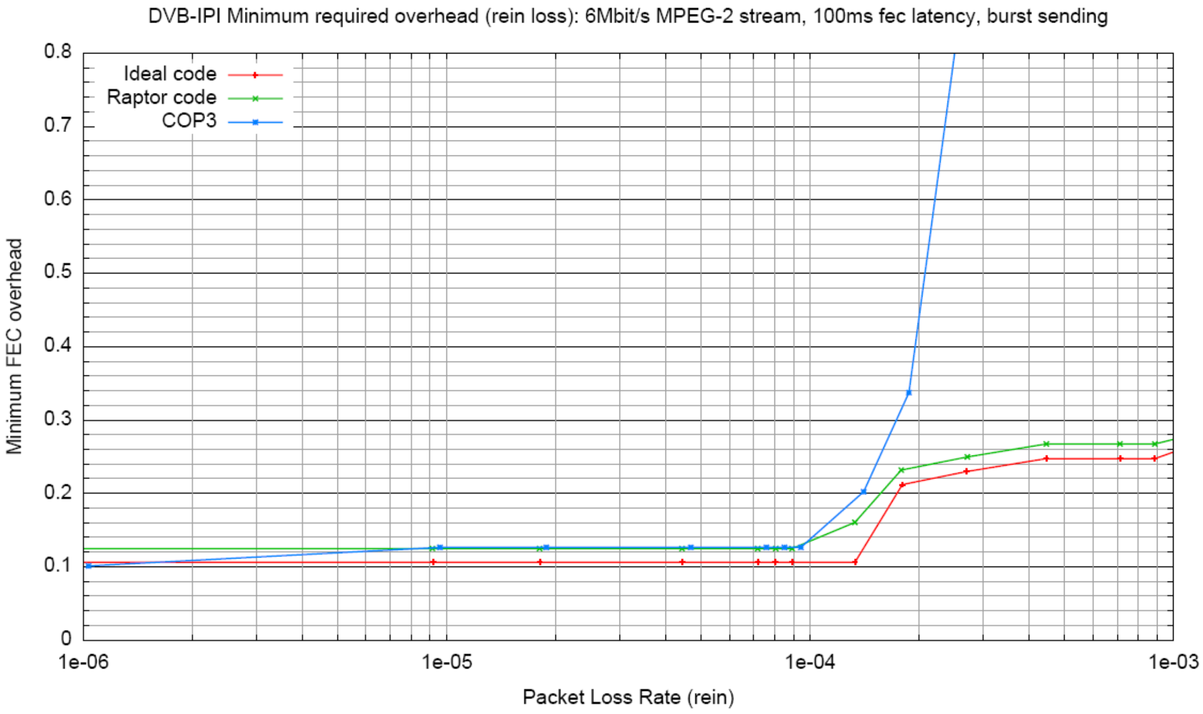

Figure A.6.3.3.2.2-4: 6 Mbit/s MPEG-2 Transport Stream, 100 ms latency (live content), REIN loss, burst sending

As in previous cases, the Raptor code meets the quality target at all error rates with overhead close to the minimum possible. The Pro-MPEG code meets the quality target with minimum overhead only in cases where the loss rate is below a threshold which is around 10^{-4} packet loss rate.

With the constant sending arrangement, and REIN losses, the Raptor codes requires an overhead which is less than or (approximately) equal to the Pro-MPEG overhead for all loss rates. For other cases (burst sending and/or random loss) the Pro-MPEG code requires marginally less overhead for the loss rates which are below the threshold.

For low loss rates and in the presence of random loss, the ProMPEG code is simple a 1D parity code, which is well known to be ideal. In these cases ProMPEG achieves lower overhead than Raptor.

A.6.3.4 A note on latency, jitter and traffic shaping

All the above simulations assume that the sent traffic should maintain a constant bit-rate (although it is accepted that the constant-bitrate ProMPEG scheme actually doubles the instantaneous bit-rate each time a repair packet is sent, this is only visible as a variation in bit-rate over very short time periods. However for the burst sending arrangement, the variation is significant and over a longer period of time).

In order to support legacy receivers in the case of multicast, whenever this is feasible, the use of FEC should not introduce significant additional jitter in the source packets. Using the sending arrangement proposed for Raptor codes does introduce a small amount of additional jitter to the arrival of source packets at the receiver. Using the constant sending arrangement proposed for Pro-MPEG avoids such jitter, however using the burst sending arrangement proposed for Pro-MPEG will introduce a small amount of additional jitter as the bursts are traffic shaped on the access link. Sending arrangements are interchangeable between the codes, so there are many possibilities. See clauses A.3 for more details. Clause A.7 gives details of the sending arrangements used in the simulations.

In the simulations above, the maximum additional jitter in the case of Raptor is around 40 ms for the 400 ms latency cases and in most cases significantly less. Finally, "latency" in these simulations has been interpreted as the additional latency introduced between the source and the playout due to the use of FEC. This is equivalent to the size of the FEC data buffer assumed to exist at the receiver. This latency adds directly to the response time for user actions, such as channel change, re-wind, forward-wind etc.

In the case of live content, the Raptor scheme as proposed adds a small additional amount to the time between the event actually occurring at the sender and the presentation to the user (distinct from the response time for user actions, referred to above). In the cases above this is at most around 40 ms and in general considerably less. Since the overall end-to-end delay is general much higher than 40 ms, this additional delay is not considered significant, especially since it does not contribute to the response time for user actions. The Raptor scheme is sufficiently flexible that this delay could be reduced if required. Targets on this end-to-end delivery time have not been discussed and again could be included in a further phase of this evaluation if necessary, but again it is unlikely to significantly affect the results.

Finally, the only two latency figures (100 ms and 400 ms) were tested in these evaluations. It is instructive to consider the trade-off involved in selection of an FEC latency figure. Lower latency results in shorter channel change time but has a cost in that a higher FEC overhead is required for a given level of protection. Conversely, a longer latency budget results in longer channel change time in return for a lower FEC overhead. Figure A.6.3.4 illustrates this trade-off for an "ideal" code and for several quality targets ("Mean Time Between Artifacts"). Figure A.6.3.4 suggests that a significant bandwidth saving is available if the latency budget is increased from 100 ms to (say) 200 ms, but that there is little to be gained by increasing the latency above 400 ms. In particular, figure A.6.3.4 throws doubt on the practical validity of the 2 MBit/s, 100 ms case evaluated above: an operator who was sufficiently bandwidth-constrained to use 2 Mbit/s encoding would surely also take advantage of the FEC bandwidth savings that could be achieved with a 200 ms latency budget.

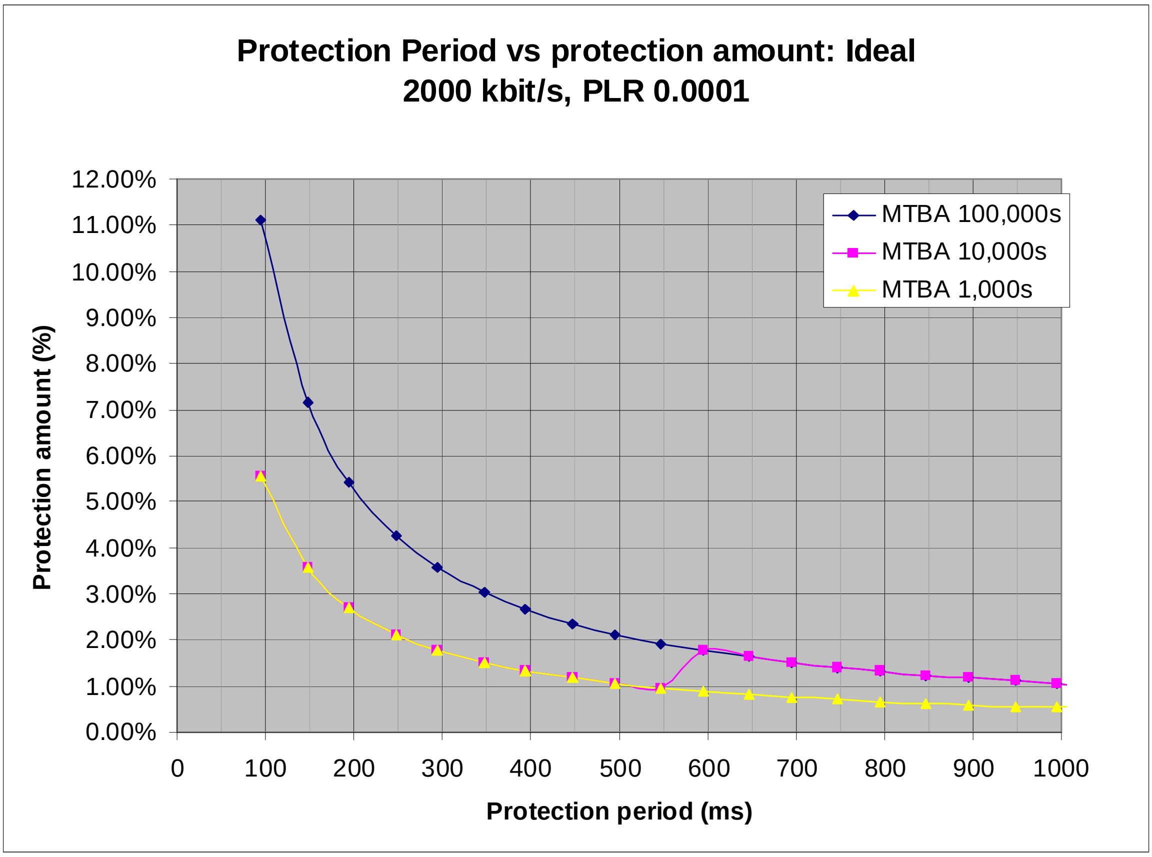

Figure A.6.3.4: Latency/FEC bandwidth trade-off

A.6.3.5 Summary of simulation results

We summarize the above results according to the sending arrangement and type of loss:

Summary for multicast and unicast live video:

- There is a "loss rate threshold" in each case: below this threshold, the Pro-MPEG overhead is very low and close to Raptor (sometimes higher, sometimes lower) and above this threshold, the Pro-MPEG overhead is significant (always much higher than Raptor overhead).
- The threshold is around 1e-4 Packet Loss Rate (actually between 5e-5 and 2e-4), depending on the case.

Constant sending arrangement, random loss:

- Below the threshold, the Pro-MPEG overhead is slightly less than the Raptor overhead and above this threshold, the Raptor overhead is much less than the Pro-MPEG overhead.

Constant sending arrangement burst (REIN) loss:

- Below the threshold, the Raptor overhead is slightly less than the Pro-MPEG overhead and above this threshold, the Raptor overhead is much less than the Pro-MPEG overhead. Please note that in this case, Raptor overhead is always the lowest.

Burst sending arrangement, random loss:

- Burst sending does not have much effect on results below the threshold.
- The Pro-MPEG overhead is reduced above the "threshold" compared to constant sending arrangement, but is still much greater than the Raptor threshold.
- Burst sending does not have much effect on the Raptor overhead.

Burst sending arrangement, burst (REIN) loss:

- The Pro-MPEG overhead is reduced both above and below the "threshold", but above the threshold is still much greater than the Raptor threshold.
- Below the threshold the Pro-MPEG overhead is slightly less than the Raptor overhead.
- Burst sending does not have much effect on the Raptor overhead.

Summary for unicast stored or buffered content:

- In the particular case of unicast stored or buffered content, Raptor code can use the faststart sending arrangement so as to use significantly less bandwidth than Pro-MPEG in all cases.
- When faststart mechanism is not used, results are the same as multicast and unicast live video.

In all cases, the results plotted above show the overhead required by the "best" configuration parameters for the Pro-MPEG COP3 code according to guidelines for setting Pro-MPEG parameters and the specification in [6]. These were chosen by searching through the various possible configurations (including row packets only, column packets only, both row and column packets and different matrix sizes) and reporting only the lowest overhead which achieved the required quality. This means that the choice of code was based implicitly on complete knowledge of the loss rates and patterns in each case.

In summary, the requirements on network quality (target end-to-end loss rates) depend significantly on the choice of FEC code (Pro-MPEG or Raptor): network quality requirements are much more stringent if Pro-MPEG is chosen since it works well only as long as the packet loss rate remains under the previously defined threshold (around 1e-4).

A.6.4 Flexibility

The FEC evaluation criteria for flexibility states:

"Flexibility:

- Changing the overhead or/and the block size dynamically (within or between FEC blocks).
- Range of protection periods.
- Suitability for use with a wide variety of FEC sending strategies'.

The Raptor code provides complete flexibility in terms of overhead (protection amount) and block size (protection period). These parameters can be set independently according to application requirements and the error correction performance of the code remains just as close to "ideal" whatever the parameter settings. Parameter settings can easily be changed dynamically and protection periods from 10 s to 1 000 s of milliseconds can be efficiently supported.

For the Pro-MPEG code, the protection period and protection amount are related and constrained and in practice only certain combinations are supported Nevertheless, the possible number of combinations is large enough to offer many different levels of protections.

A.6.5 Processing and Memory requirements

The Raptor code has been designed to have very modest computational complexity such that it is easy to implement in software on resource constrained devices such as Set-Top Boxes and mobile devices. Techniques for efficient hardware implementation for high capacity encoders have also been presented and many options exist for hardware-assisted implementations for decoders.

The Pro-MPEG code has been designed to have very low computational complexity such that it is easy to implement it in software or in hardware.

For both Raptor and Pro-MPEG, the complexity of encoding is comparable with the complexity of decoding. For Raptor, both scale linearly with the volume of data to be encoded/decoded, making the overall computational requirements proportional the service bit-rate and to a large extent independent of the losses or level of protection.

Raptor encoding complexity for the scenarios considered here is in the region of 2 MIPS per Mbit/s - so a 6 Mbit/s stream would require ~12 MIPS of processing power to encode, although in practice the encode time is also dependent on memory bus speed and cache/DMA availability. For example, Digital Fountain has demonstrated an off-the-shelf rack-mounted server with a Pentium processor running at 3 GHz performing Raptor encoding at 2 Gbit/s - the equivalent of 1,000 2 Mbit/s video streams. Further optimizations for the specific case of video stream encoding and platform-specific optimization could be expected to increase this encoding speed significantly. Leading Pro-MPEG COP3 processing cards encode at around 400 Mbit/s and so similar performance could be easily achieved with Raptor with modest processing requirements.

Hardware optimizations of Raptor codes in the form of hardware assist for XOR operations or complete implementation of the code in hardware are also possible and can further improve capacity. The application of the Raptor code for streaming has been designed so that for a given stream rate/latency the block size and structure from the encoders point of view is the same for every block. Thus the sequence of operations required to encode repair packets for a block can be calculated or stored in advance and executed quickly (in software or hardware) for each block. This is true even if the actual block size (in terms of packets) differs between protection periods.

The number of primitive symbol XOR operations required for Raptor encoding or decoding for the scenarios considered here is around 12 to 14 operations for each source symbol.

The number of primitive symbol XOR operations required for Pro-MPEG encoding or decoding for the scenarios considered here is 1 operation for each source symbol in Pro-MPEG 1D and 2 operations for each source symbol in Pro-MPEG 2D.

Nevertheless, in practice, for each symbol, these operations are performed on-chip (in cache) and so the bottleneck is the speed with which data can be moved between memory and the processor, rather than the precise number of XOR operations. All modern processors employ pipelining and so can perform the XOR operations on-chip concurrently with moving data for future operations between off- and on-chip memory. This means a reduction in XOR operations does not necessarily translate into a significant increase in speed of encoding or decoding.

With Raptor, minimum memory requirements for data to be encoded/decoded at both encoder and decoder are slightly greater than the source block size. At the decoder, received data (which is a mix of source data and repair data) may be transformed "in-place" into the recovered source block. Thus, these memory requirements are less than 350 KB for the largest block size considered in this evaluation.

With Pro-MPEG, the encoder only needs to have buffers so as to store the repair packets of a protection block. Since amount of protection is always much lower than the amount of data, it means a Pro-MPEG encoder requires memory much smaller than the source block size. On the decoder, Pro-MPEG only requires enough memory to store the current protection block and its repair packets. Therefore it means a Pro-MPEG decoder requires memory slightly greater than the source block size. Note also, that depending on the sequencing arrangement used, the decoder may need more memory. For instance, when repair packets are arranged within the block after the one they protect, the decoder would need twice as much memory to store the current and following protection blocks.

Note that for decoders, this memory requirement is still very modest compared to the memory required, for example, for storing a single HD frame after decoding.

A.6.6 Additional criteria

The following additional criteria are included in the evaluation criteria document:

- Continued functioning of existing STB products in presence of FEC data.
- Option for new STB products to use or ignore FEC data.
- Confirmation of FEC scheme IPR compliance with DVB rules.
- Support of combined protection of different streams (such as when audio and video packets are sent in two separate streams).

Raptor is compliant to all these criteria.

Pro-MPEG is compliant to the first two criteria and believed to be compliant to the third (IPR compliance is currently being clarified by SMPTE).

The Pro-MPEG code does not support combined protection of different streams - separate protection streams are required for each RTP flow. Specifically in the case of audio streams, which have much lower bandwidth than the video streams, then high quality protection will be extremely difficult to achieve if latency needs to be kept very small.

In general, combined protection is more efficient than separate protection and in particular separate protection of the relatively low bit-rate audio stream can be extremely inefficient.

Combined protection can also encompass the RTCP packets that provide time synchronization information between the audio and video streams.

A.6.7 Content Download

It has been suggested that the FEC solution chosen for streaming services should also be suitable for use in content download applications. It should be noted that it has not yet been agreed, (or even discussed in detail), that Forward Error Correction is required for Content download - other solutions do exist. An evaluation of these solutions should be carried out by the TM-IPI Content Download System (CDS) taskforce.

However, solutions based on forward error correction have a number of significant advantages over other solutions in the multicast case. The Raptor code proposed for DVB-IP streaming applications is highly suitable for content download applications as well (and has been adopted for such applications by 3GPP and DVB CBMS). The same code could therefore be used for both streaming and content download.

No description is available of whether and how the Pro-MPEG code could be applied to content downloading: it was clearly designed for streaming services in extremely low packet loss cases only. The Pro-MPEG code is by nature a short block code and for content downloading a large block code is much more efficient if FEC is to be used.

A.6.8 Raptor vs. Pro-MPEG Summary

The table below summarizes the results described above. The green font identifies the best result while the red font identifies the worst result. When the result between codes is very close, an orange font is used to identify the code that only performs slightly less well.

Criteria	Pro-MPEG Constant	Pro-MPEG Burst	Raptor	Comments
Bandwidth cost - loss rates > ~1e-4				
- SD MPEG-2 TS broadcast (400 ms)	High	High	Low	
- HD MPEG-2 TS broadcast (400 ms)	High	High	Low	
- SD MPEG-2 TS unicast (100 ms)	High	High	Low	Thanks to its fast-start mechanism, Raptor achieves very low overhead in case of stored/buffered content
- HD MPEG-2 TS unicast (100 ms)	High	High	Low	Thanks to its fast-start mechanism, Raptor achieves very low overhead in case of stored/buffered content
Bandwidth cost - loss rates < ~1e-4				
- SD MPEG-2 TS broadcast (400 ms)	Low	Lowest	Low	
- HD MPEG-2 TS broadcast (400 ms)	Low	Lowest	Low	
- SD MPEG-2 TS unicast (100 ms)	Modest	Lowest	Low	Thanks to its fast-start mechanism, Raptor achieves very low overhead achieved in case of stored/buffered content
- HD MPEG-2 TS unicast (100 ms)	Modest	Lowest	Low	Thanks to its fast-start mechanism, Raptor achieves very low overhead achieved in case of stored/buffered content
Support of target quality for evaluated packet loss range/patterns	See comment		Yes	Pro-MPEG COP3 could not provide a Mean Time Between Packet Loss of 4 hours for a number of the burst loss cases. However, a slightly weaker target of Mean Time Between Artifacts (visible errors) of 4 hours could be achieved.
Further packet losses that could occur in the core network due to congestion and/or the home environment e.g. wireless technologies.	-		-	Not yet evaluated
Flexible engineering of code parameters	Yes (but fixed number of combinations and direct correlation between overhead and protection block size)		Yes (fully)	
Computational complexity	Lowest		Modest	
Scalability (e.g. encoding of 1 000 s of streams)	Yes		Yes	
Memory requirements (encoder)	Lowest		Modest	
Memory requirements (decoder)	Modest		Modest	
Visibility of artifacts after FEC decoding	-		-	Both codes could perform partial correction.
Continued functioning of existing STB products in presence of FEC data	Yes		Yes	
Option for new STB products to use or ignore FEC data	Yes		Yes	
Confirmation of FEC scheme IPR compliance with DVB rules	Yes		Yes	Pro-MPEG IPR compliance is currently

Criteria	Pro-MPEG Constant	Pro-MPEG Burst	Raptor	Comments
				under SMPTE process.
Efficient support of direct encapsulation of audio/video in RTP (as defined in TS 102 005 [10]): Support of combined protection of audio and video packets	No		Yes	Raptor can protect several RTP and RTCP streams together whereas Pro-MPEG has to consider each RTP and RTCP streams separately.
Efficient support of direct encapsulation of audio/video in RTP (as defined in TS 102 005 [10]): support of variable length packets	Yes (but less efficient)		Yes	
Suitable for Content Download Service	No (much less efficient)		Yes	

A.6.9 Conclusions

The sending arrangement chosen has a significant impact on the performance / bandwidth cost.

The comparison of the two codes also differs depending on the packet loss rate.

In the case that burst sending is used and for loss rates below a threshold (between 5e-5 and 2e-4), the Pro-MPEG code requires slightly less bandwidth than Raptor code.

In the case that burst sending is not used and for loss rates below a threshold (between 5e-5 and 2e-4), both Pro-MPEG and Raptor codes requires similar bandwidth overhead although there are differences depending on the precise case (see clause A.6.3.5).

For loss rates above a threshold (between 5e-5 and 2e-4), Raptor code requires much less bandwidth than Pro-MPEG code.

The threshold indentified through these simulations depends on quality target, source stream bitrate, latency budget and loss patterns.

When the Raptor fast-start mechanism is used for unicast/buffered content, Raptor requires less overhead than Pro-MPEG.

Regarding implementation aspects (complexity, memory requirements, etc.), though there are differences between codes (see clause A.6.8), no significant issues were identified with either code.

Both codes meet the requirement for backward compatibility with existing equipments.

The Raptor code supports various future requirements which the Pro-MPEG does not (see clause 9).

Since neither of these two codes is optimal in all cases, an hybrid code with performance similar to the best of either was defined (see clause A.9 for simulation results).

A.7 Sending arrangements used for simulations

A.7.1 DF Raptor default sending arrangement

The sending arrangement proposed for the DF Raptor code is illustrated in figure A.7.1. In this sending arrangement the overall sending rate is kept constant and the source packets of each block are sent before any of the repair packets of the block. This approach requires that the sending rate of the source packets be increased marginally to make space for the repair packets at the end of the block.

It is important to note that the sequencing of packets is determined by the FEC procedures which operate "below" the RTP layer. The contents of the packets, in particular the RTP timestamps, are not modified compared to the contents in the case in which FEC is not applied and therefore the correct timing for the packets can be reconstructed with the usual procedures.

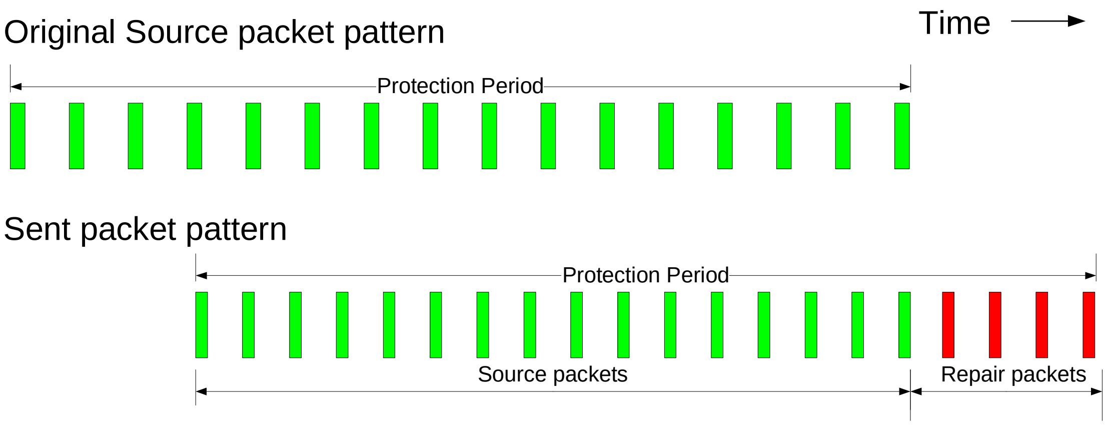

Figure A.7.1: DF Raptor sending arrangement

Note that while this arrangement ensures a global constant bitrate, it actually modifies the rate at which source packets are sent and consequently creates a small amount of additional jitter on the transmission.

Other sending arrangements are also possible for DF Raptor but were not investigated.

Pros and cons:

+ global sending rate is constant.

+ full latency budget available for FEC protection.

- source data sending rate is different from original source data sending rate.

- insertion of repair packets introduces small amount of jitter on all source packets.

A.7.2 Pro-MPEG COP3 fully interleaved sending arrangement

Annex C of the Pro-MPEG specification proposes a sending arrangement as illustrated in figure A.7.2. In this sending arrangement the overall sending rate is kept constant and the sending rate of source packets is also kept constant.

Because this sending arrangement distributes repair packets for one block over the entire duration of the next block, then the maximum block size is limited to one half of the latency budget. As a result, the overhead required by the code is increased. This is illustrated in the "constant sending arrangement" results above.

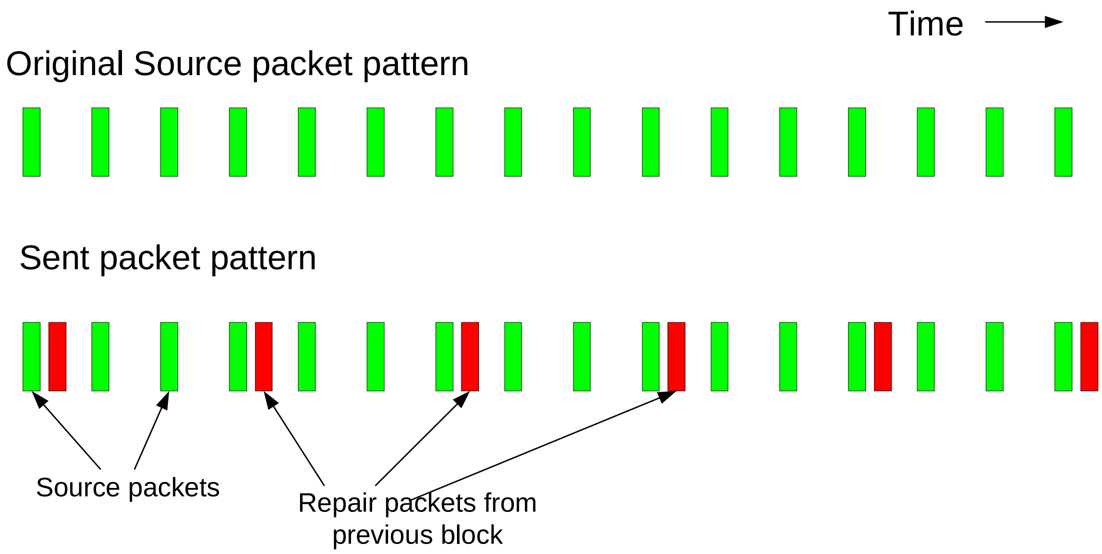
Time
Original Source packet pattern
Sent packet pattern
Source packets
Repair packets from previous block

Figure A.7.2: Pro-MPEG COP3 fully interleaved sending arrangement

Pros and cons:

- + source data sending rate is the same as original source data sending rate.
- + global sending rate is kept constant.
- - only half of latency budget is available for FEC protection.
- - insertion of repair packets introduces very small amount of jitter at the beginning when total stream bandwidth is close to available channel bandwidth.

A.7.3 Pro-MPEG COP3 burst sending arrangement

This arrangement is illustrated in figure A.7.3. In this case, repair packets for one block are interleaved with the first few packets of the next block. As a result, the instantaneous sending rate during these first few packets is significantly increased. However, the block size may now be set almost as large as the latency budget, which reduces the required overhead. This is illustrated in the "burst sending" results above.

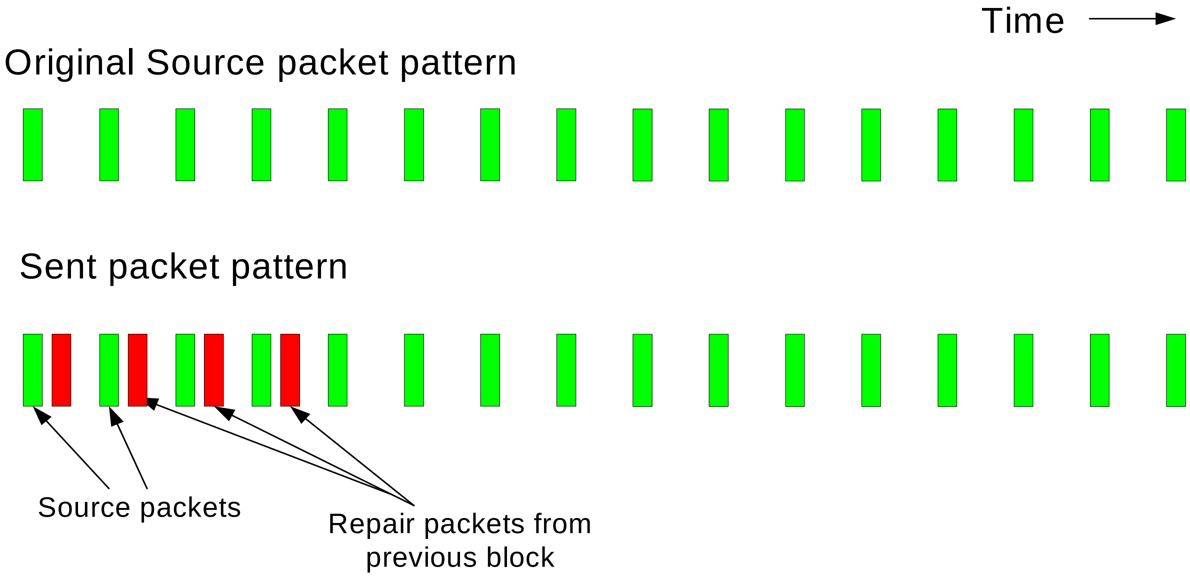
Time
Original Source packet pattern
Sent packet pattern
Source packets
Repair packets from previous block

Figure A.7.3: Pro-MPEG COP3 burst sending arrangement

Pros and cons:

- + source data sending rate is the same as original source data sending rate.
- + almost all of latency budget is available for FEC protection.
- - global sending rate is very bursty (and therefore not constant).
- - insertion of repair packets introduces small amount of jitter at the beginning when total stream bandwidth is close to available channel bandwidth.

A.7.4 Concurrent Interleaved sending

In the case of Video on Demand, or if additional latency at the encoder is acceptable, a sending arrangement as depicted in figure A.7.4 is possible. In this case, repair packets are interleaved within the block that they protect. This is possible in the Video on demand case because the data to be protected is available for FEC calculations to be performed slightly in advance of sending the data. Alternatively, a live stream can be buffered at the encoder for long enough for the FEC calculations to be performed before beginning to send the source packets of the block.

This sending arrangement could also be used for live content with a penalty that buffering equal to the block size would be required at the sender. This buffering contributes additional end-to-end delay to the playout of live streams i.e. the delay between a live event occurring and being presented on the user's screen. However it would not contribute additional channel change delay. This option may be important if there is existing equipment which is affected by changes in the timing of source packets. The procedures for timing recovery specified in TS 102 034 [1], annex A allow MPEG 2 timing to be recovered even in the presence of significant IP packet arrival jitter - however, if these procedures have not been correctly implemented then equipment may be adversely affected by the additional jitter introduced by some of the other sending arrangements described here.

This sending arrangement has the desirable properties that both the source packet data rate and the total data rate are constant. However, in the Pro-MPEG case, unlike the constant data rate arrangement in A.7.2, the whole latency budget can be used for a single source block.

New simulation results are presented for this sending arrangement in clause A.8. Note that only the Pro-MPEG column code was tested, not the 2D code.

For random loss, the results are similar to the comparison between Raptor with constant sending and Pro-MPEG with burst sending - i.e. Pro-MPEG uses slightly less overhead below the loss rate threshold than Raptor does. However, for burst loss, the Pro-MPEG code is significantly affected by interleaving of repair packets with the source packets they protect. For the 2 Mbit/s stream, this pushes the threshold where Pro-MPEG performs well down to 1e-5 or below. For the 6 Mbit/s stream, the quality target was not achievable: it is easy to see why, since a burst loss of 6 source packets will often hit a repair packet as well, and it is not possible with only 6 repair packets per block to avoid that the burst hits a source packet that is protected by that repair packet.

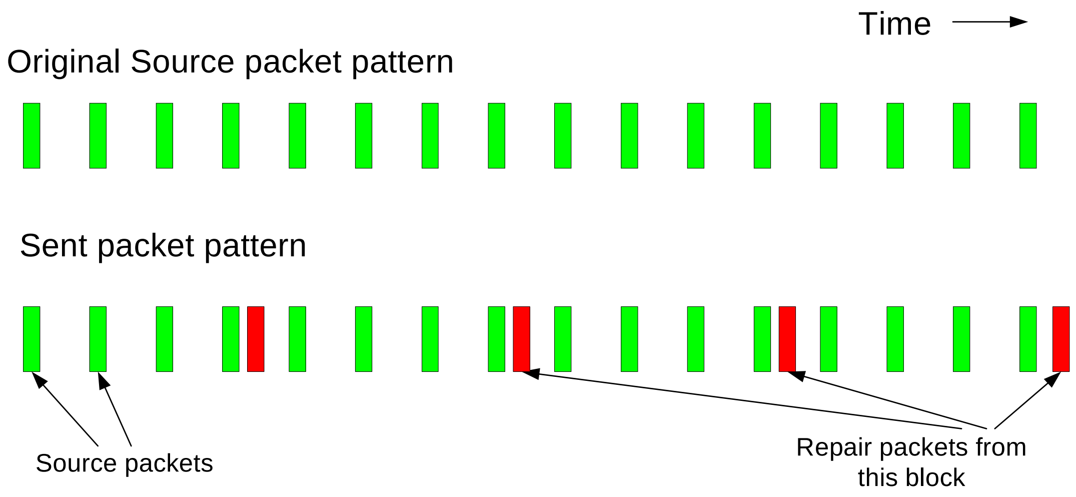
Time
Original Source packet pattern
Sent packet pattern
Source packets
Repair packets from this block

Figure A.7.4: Interleaved sending for VoD

Pros and cons:

- + source data sending rate is the same as original source data sending rate;
- + all of latency budget is available for FEC protection;
- - global sending rate is kept constant;
- - insertion of repair packets introduces very small amount of jitter at the beginning when total stream bandwidth is close to available channel bandwidth;
- - not resilient to burst losses for the Pro-MPEG FEC.

A.7.5 DF Raptor faststart sending for stored/buffered content

An additional sending arrangement for stored or buffered content (i.e. VoD and trick modes on live content) was proposed and simulated for DF Raptor. This sending arrangement is illustrated in figure A.7.5. In this arrangement, source data is sent slightly faster than the nominal stream rate at the start of the session or when trick modes are used. This allows the buffering period to be gradually increased without introducing additional channel change latency.

Two variants of this approach were simulated:

- "faststart with constant rate sending" - in which the additional source data bandwidth is obtained by reducing the FEC bandwidth at the beginning of the stream. As a result the total stream rate remains constant, but stream quality is reduced for these few initial seconds.
- "faststart with variable rate sending" - in which the overall stream rate at the beginning of the stream is somewhat higher than the nominal stream rate (e.g. 20 % higher) for the initial few seconds of the stream, but as a result the stream quality is maintained.

The second variant provided the best results.

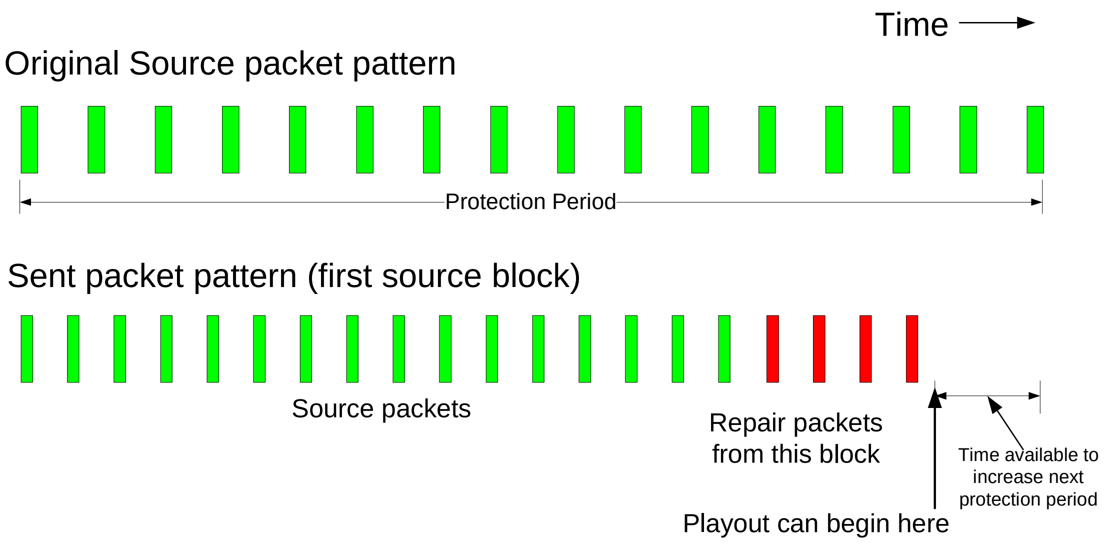
Time
Original Source packet pattern
Protection Period
Sent packet pattern (first source block)
Source packets
Repair packets from this block
Time available to increase next protection period
Playout can begin here

Figure A.7.5: DF Raptor faststart sending arrangement

Pros and cons:

- + FEC protection period can be increased to much greater than the latency budget;
- - only applicable to unicast/buffered content for Raptor.

A.8 Concurrent interleaving results

This clause presents simulation results for the sending arrangement described in A.7.4 in which both the source packet rate and the total stream rate are kept constant, whilst also allowing the full latency budget to be used for the FEC block.

Note that, due to lack of time, these results do not include the Pro-MPEG 2D code. It might be expected that in some of the cases where a result is not shown with the 1D code then the 2D code could provide the target quality, but at a relatively high overhead.

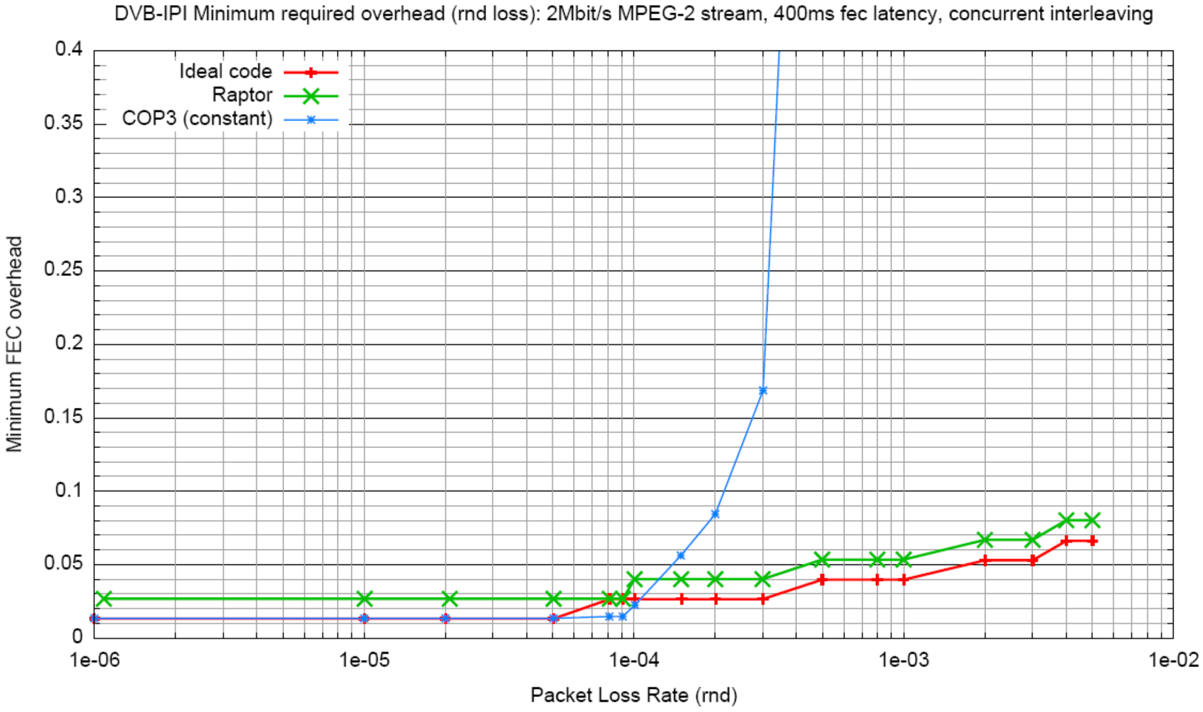

Figure A.8-1: 2 Mbit/s MPEG-2 Transport Stream, 400 ms latency, Random Loss, concurrent interleaving

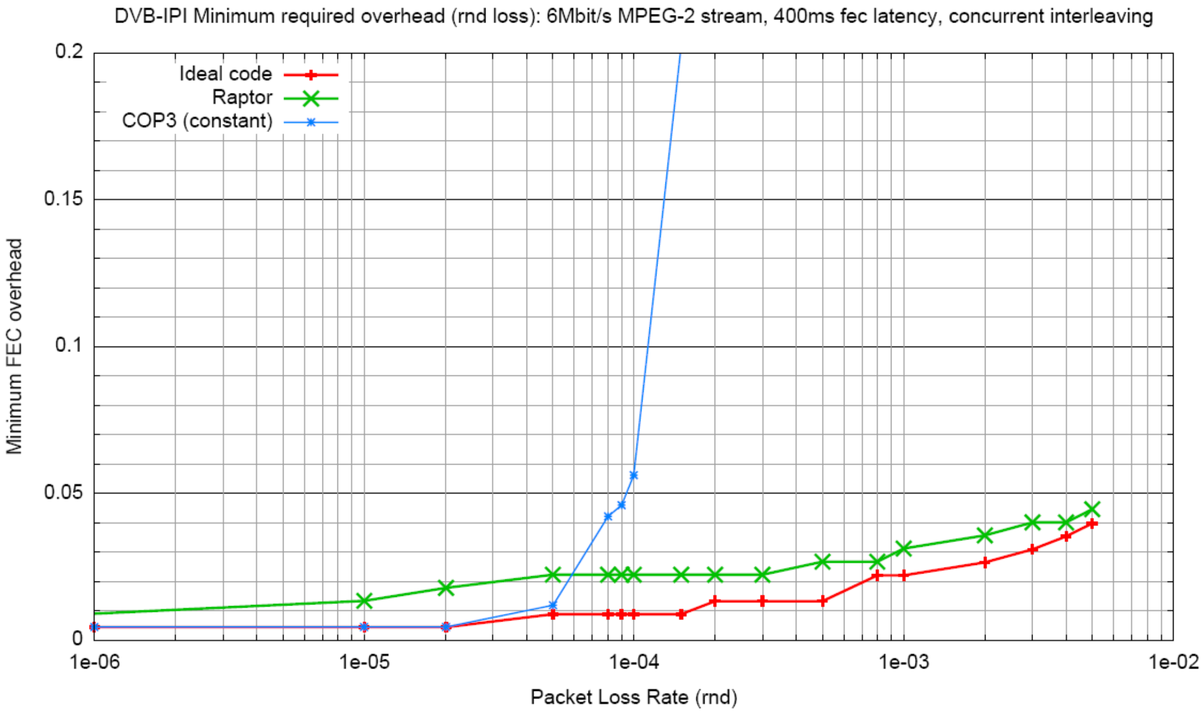

Figure A.8-2: 6 Mbit/s MPEG-2 Transport Stream, 400 ms latency, Random Loss, concurrent interleaving

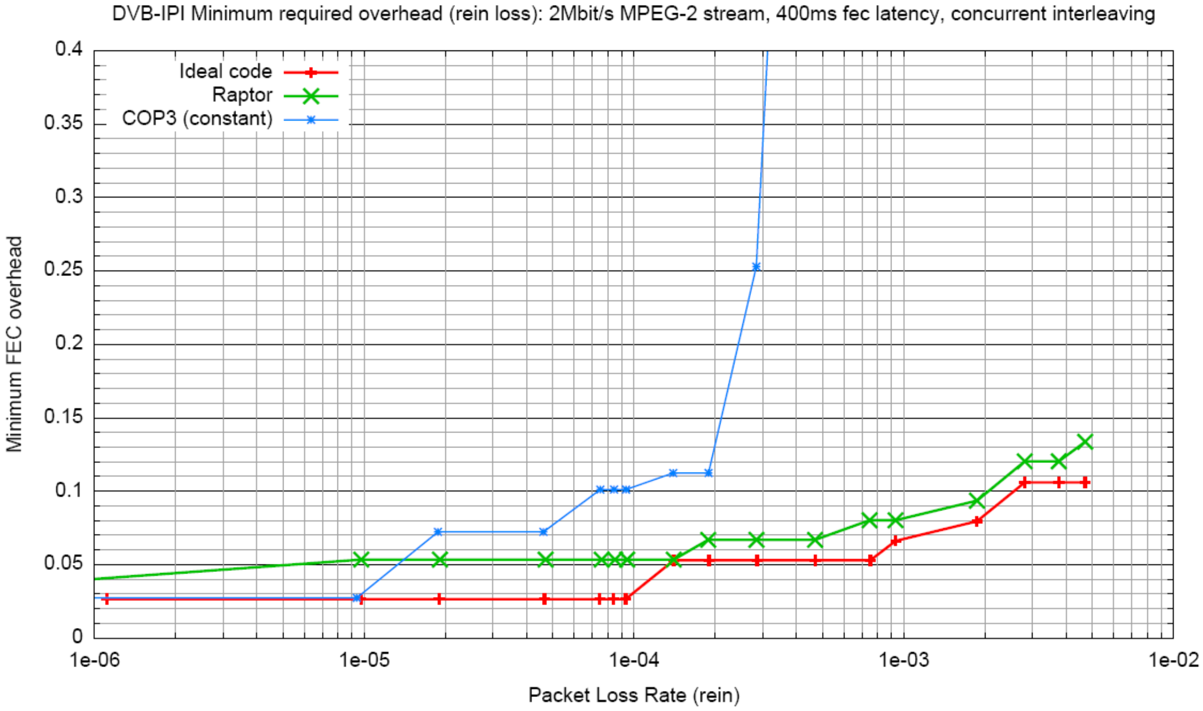

Figure A.8-3: 2 Mbit/s MPEG-2 Transport Stream, 400 ms latency, REIN Loss, concurrent interleaving

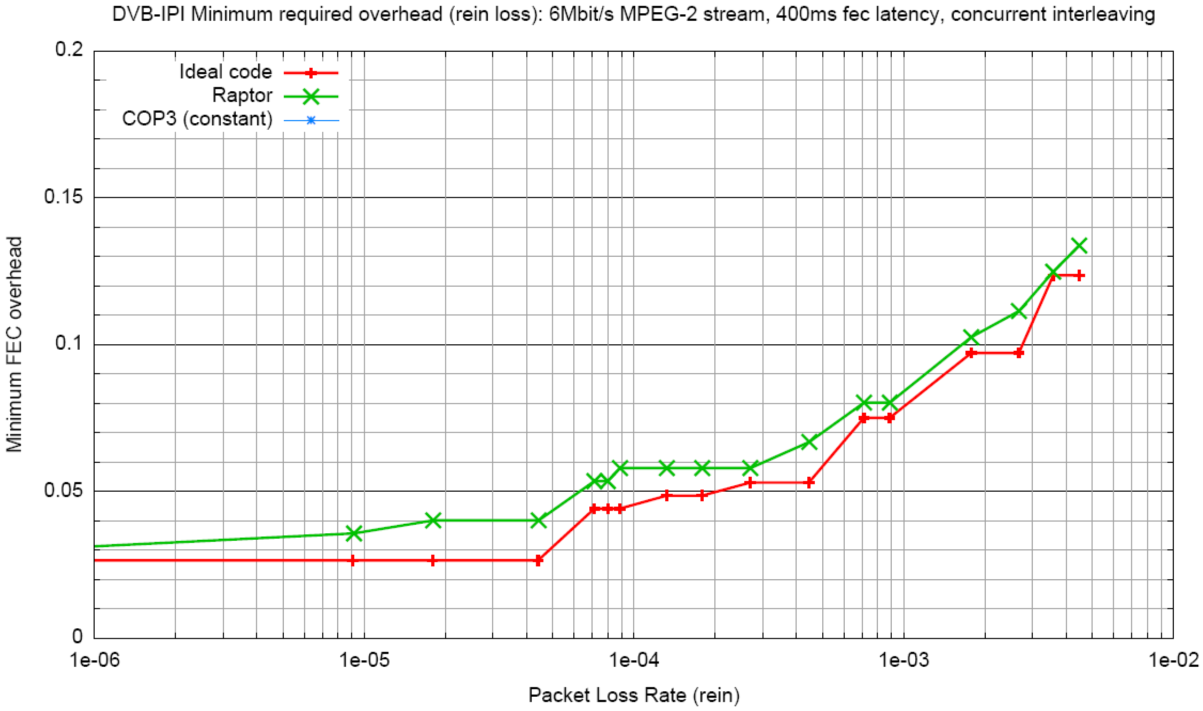

Figure A.8-4: 6 Mbit/s MPEG-2 Transport Stream, 400 ms latency, REIN Loss, concurrent interleaving

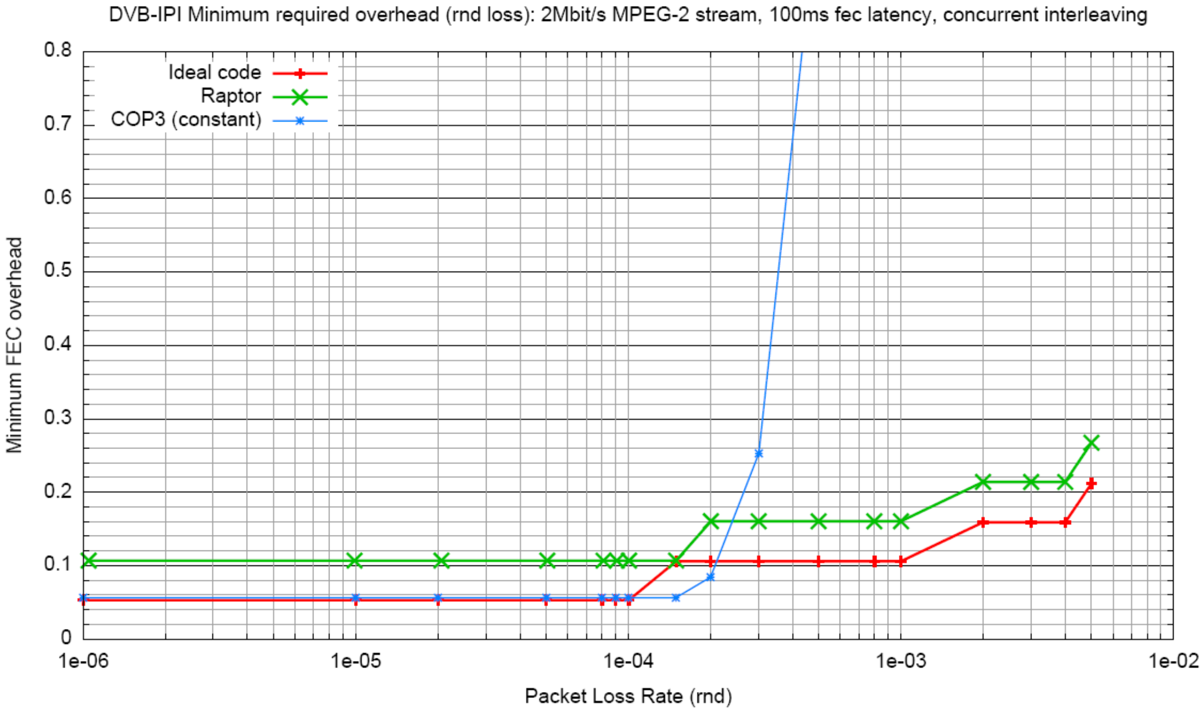

Figure A.8-5: 2 Mbit/s MPEG-2 Transport Stream, 100 ms latency, Random Loss, concurrent interleaving

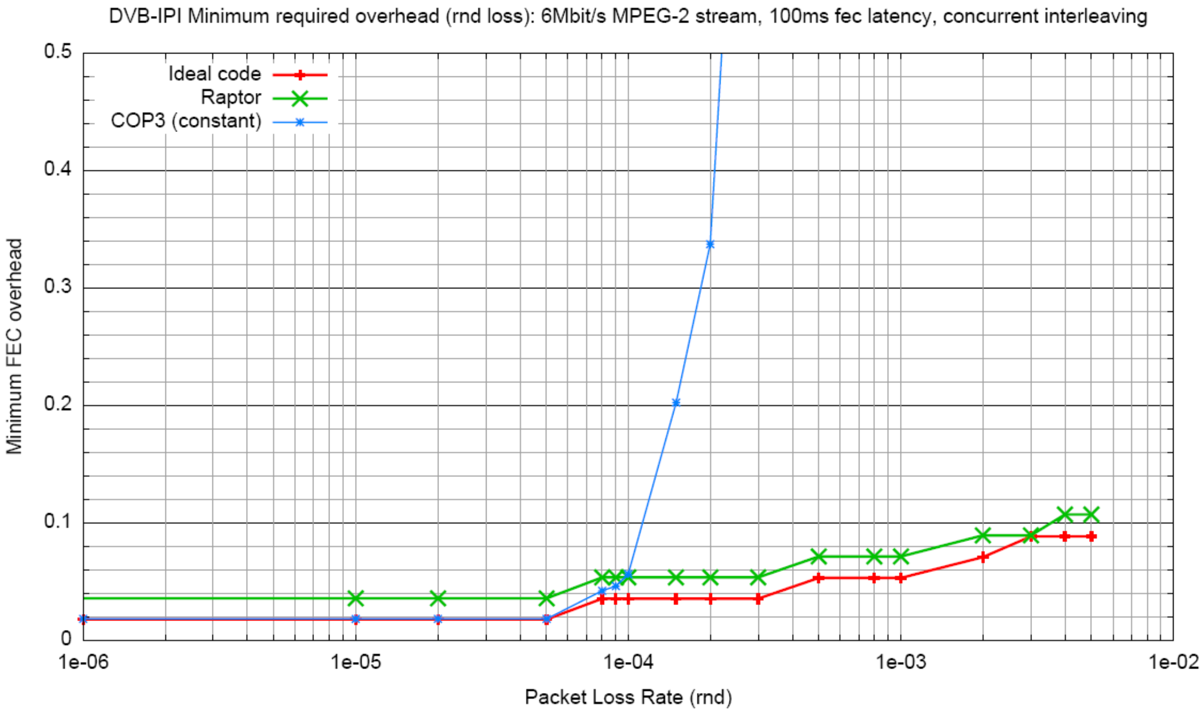

Figure A.8-6: 6 Mbit/s MPEG-2 Transport Stream, 100 ms latency, Random Loss, concurrent interleaving

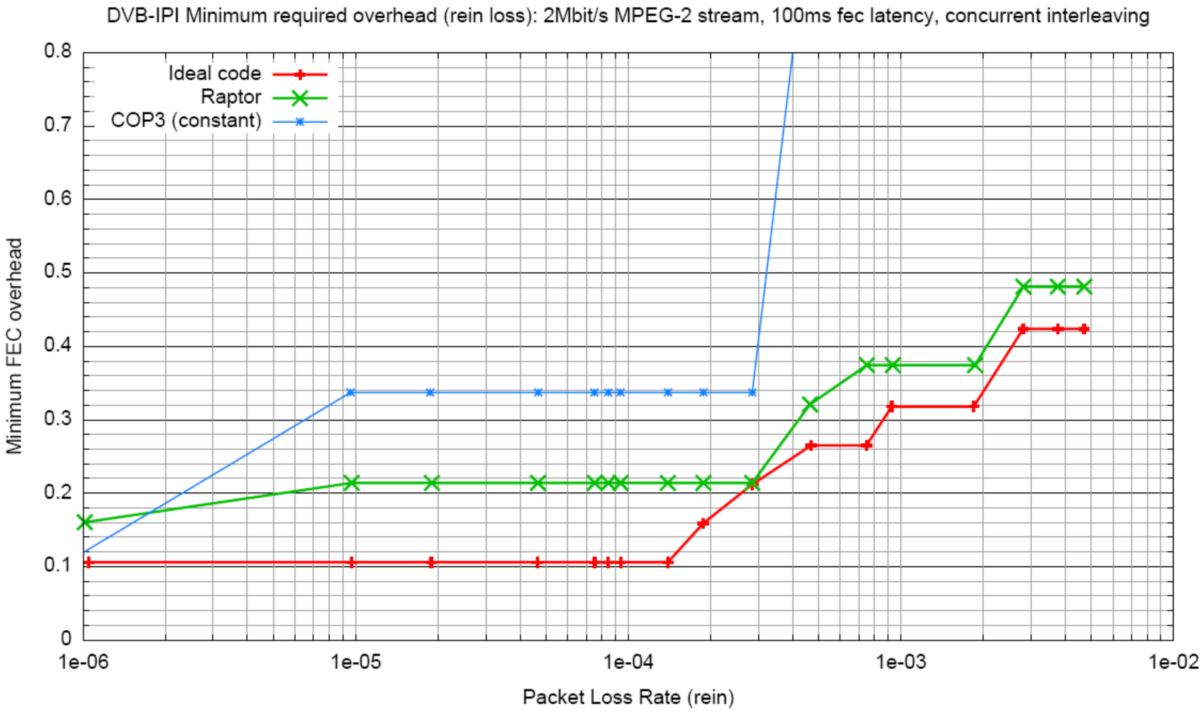

Figure A.8-7: 2 Mbit/s MPEG-2 Transport Stream, 100 ms latency, REIN Loss, concurrent interleaving

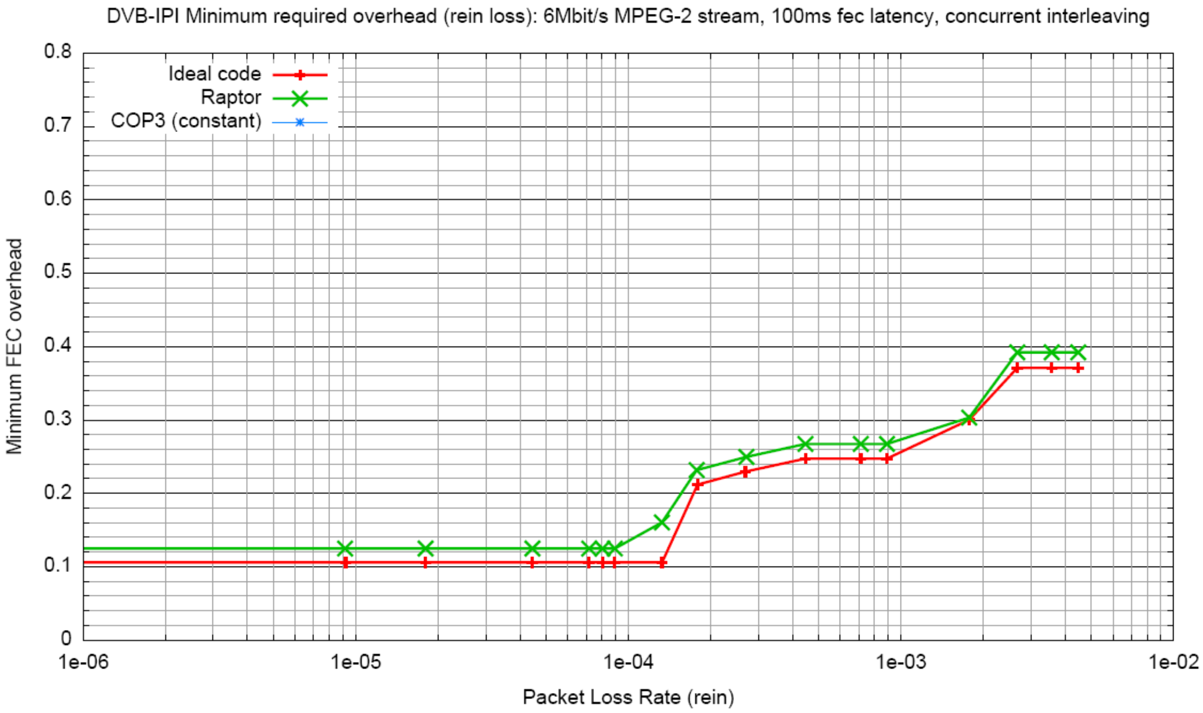

Figure A.8-8: 6 Mbit/s MPEG-2 Transport Stream, 100 ms latency, REIN Loss, concurrent interleaving

A.9 Hybrid code

A hybrid of the Pro-MPEG 1D column code and the Raptor code was proposed in order to provide a single scalable FEC solution with performance similar to the best of either the Pro-MPEG or Raptor codes in any given case.

A.9.1 Hybrid code results

This annex presents results for the Hybrid code. The hybrid cases are denoted "Raptor P<n>" where <n> is the number of parity packets used. The value of <n> chosen in each case is the smallest such that the quality target can be achieved with Pro-MPEG packets alone at loss rates of 1e-5 and lower.

The sending arrangement of clause A.7.1 was used for these simulations.

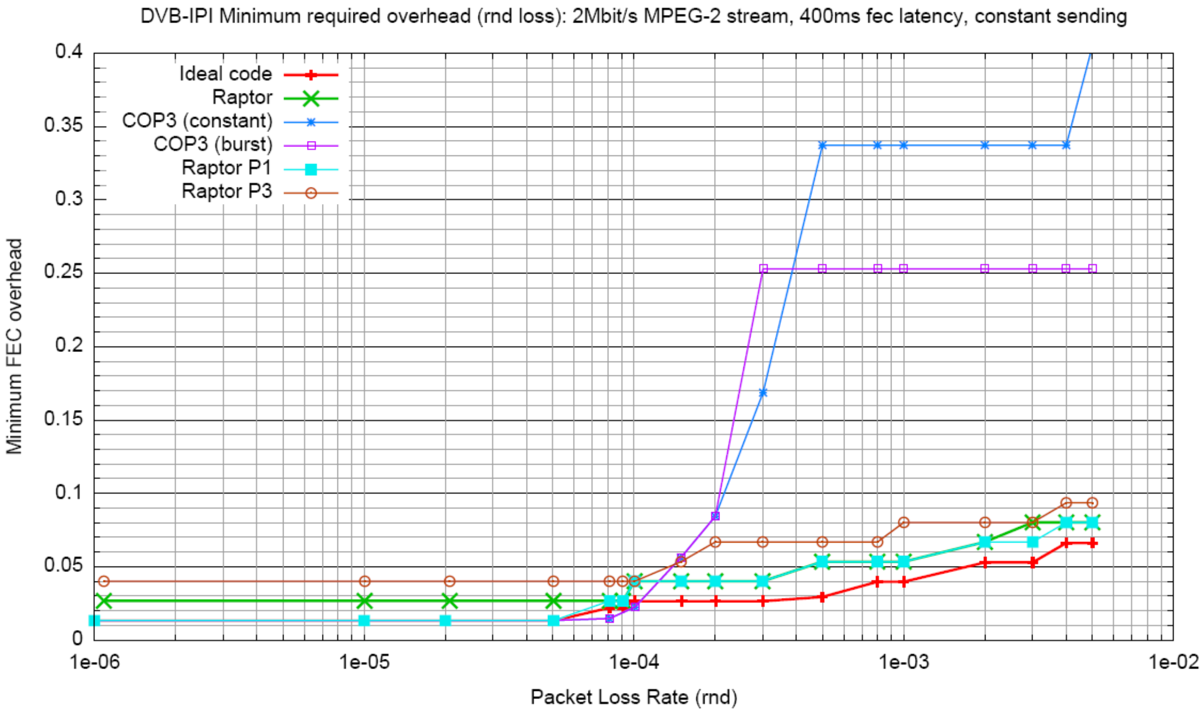

Figure A.9-1: 2 Mbit/s MPEG-2 Transport Stream, 400 ms latency, Random Loss, constant sending

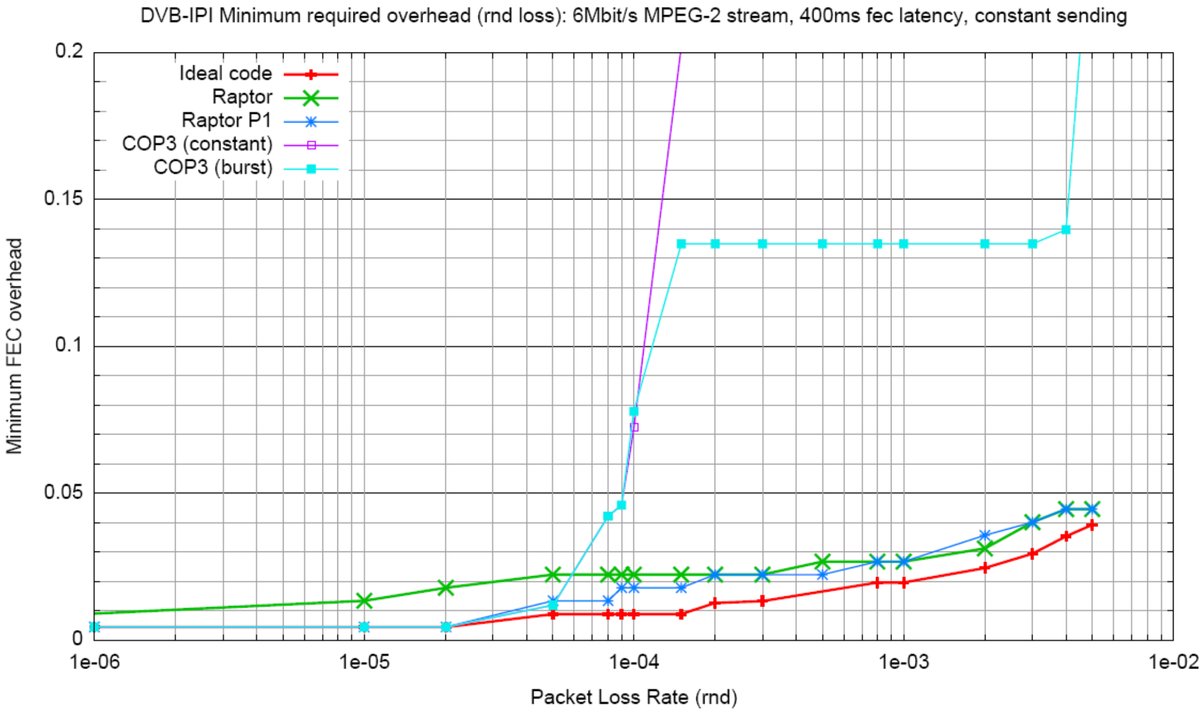

Figure A.9-2: 6 Mbit/s MPEG-2 Transport Stream, 400 ms latency, Random Loss, constant sending

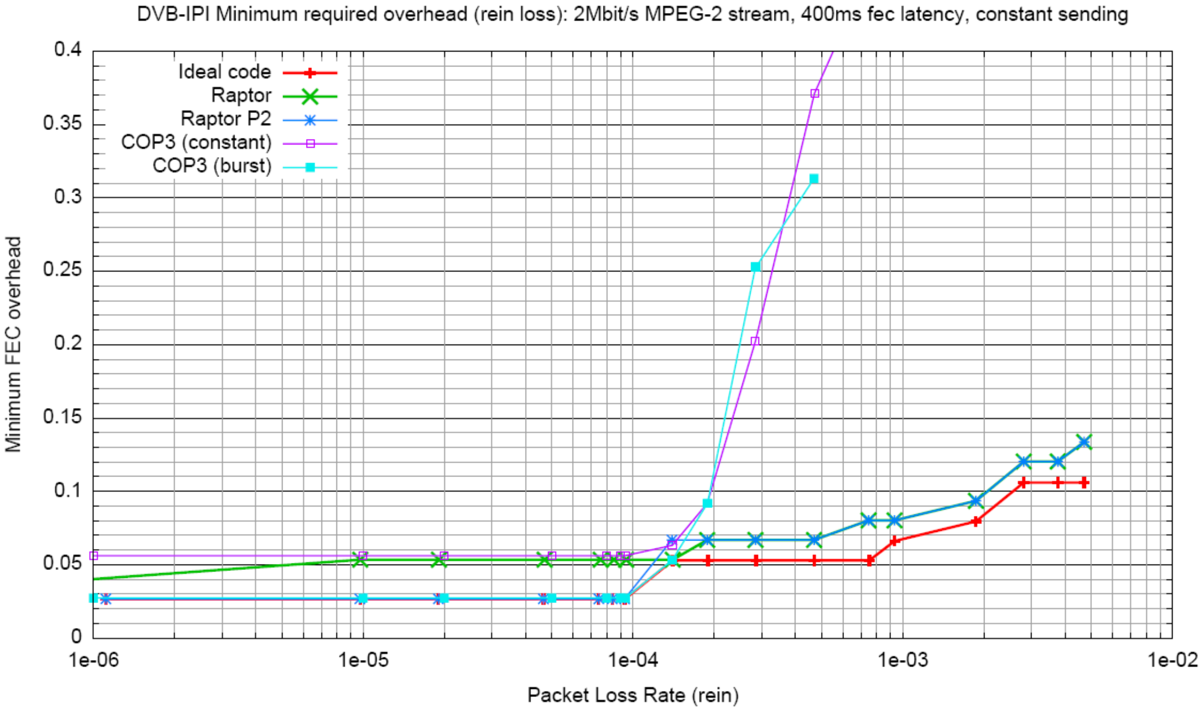

Figure A.9-3: 2 Mbit/s MPEG-2 Transport Stream, 400 ms latency, REIN Loss, constant sending

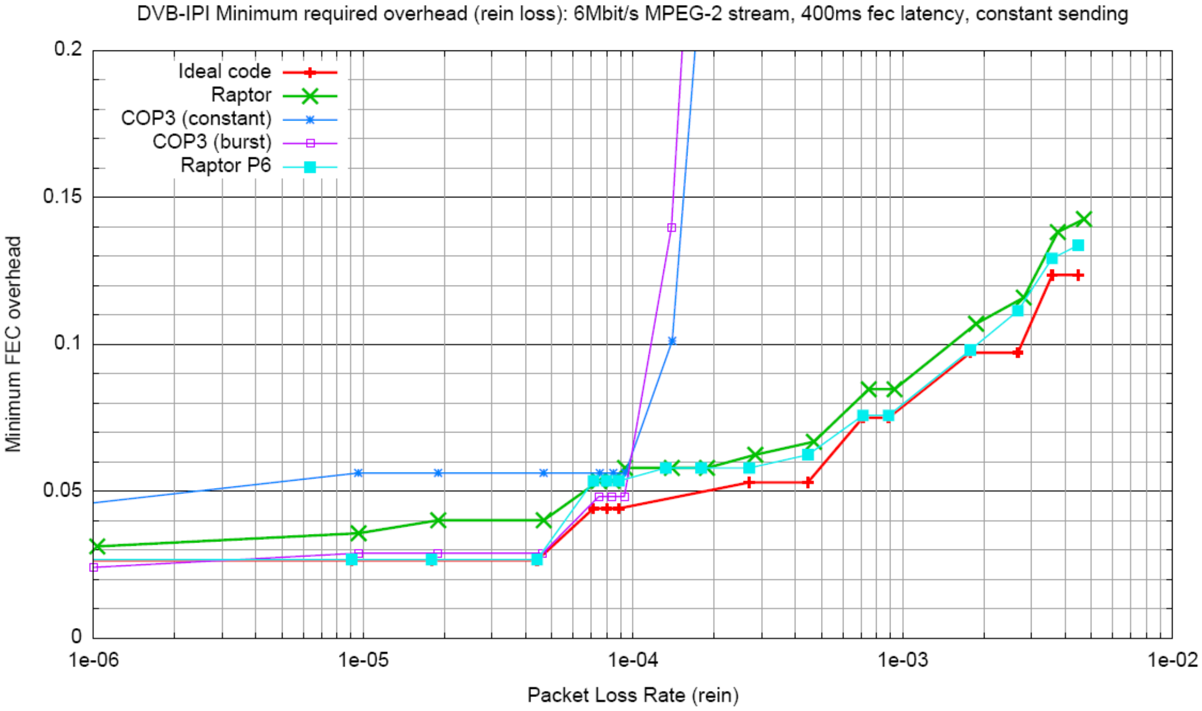

Figure A.9-4: 6 Mbit/s MPEG-2 Transport Stream, 400 ms latency, REIN Loss, constant sending

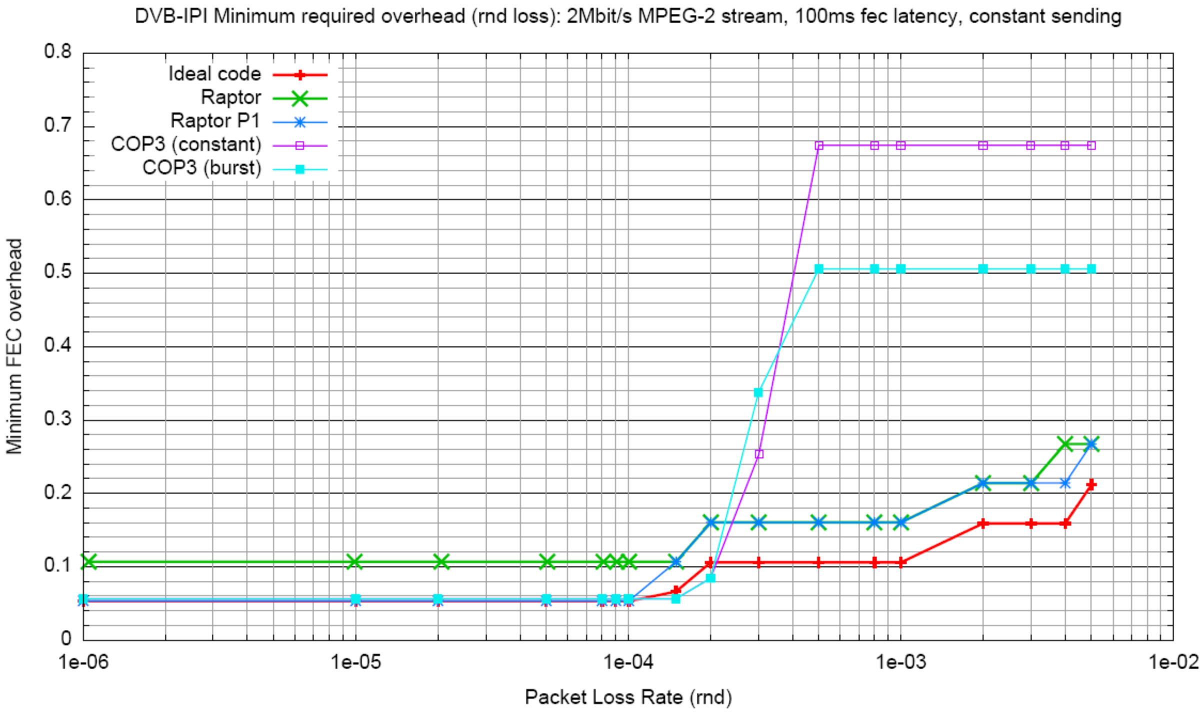

Figure A.9-5: 2 Mbit/s MPEG-2 Transport Stream, 100 ms latency, Random Loss, constant sending

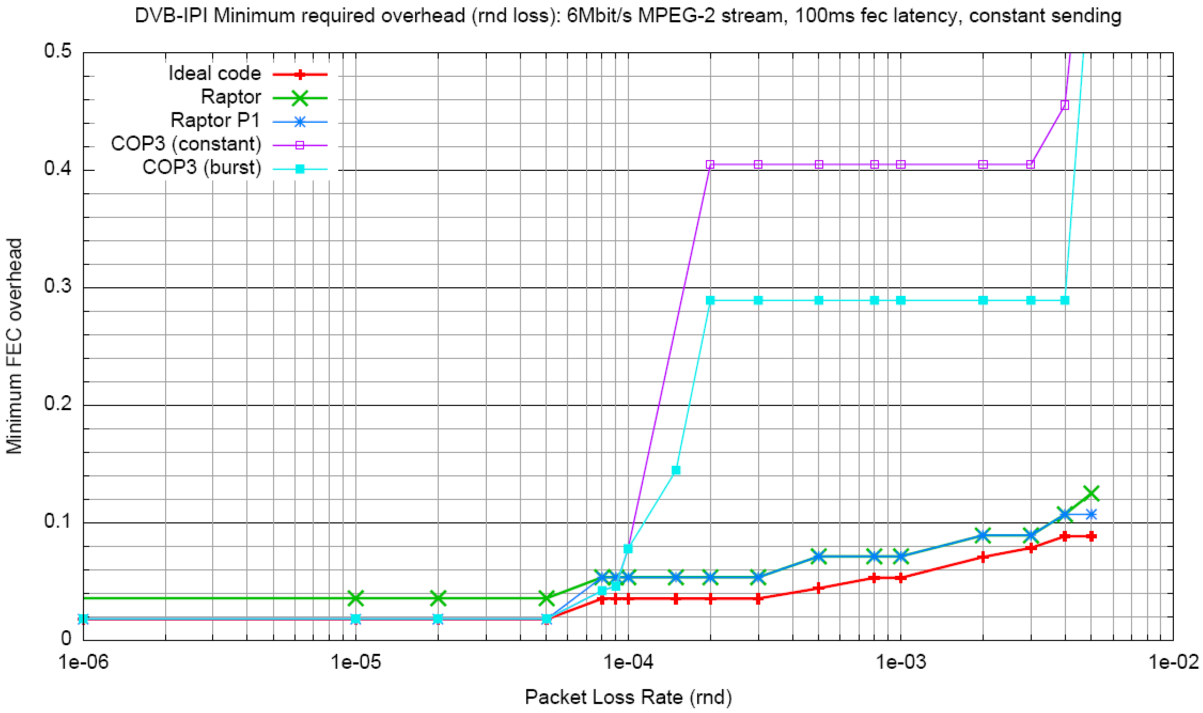

Figure A.9-6: 6 Mbit/s MPEG-2 Transport Stream, 100 ms latency, Random Loss, constant sending

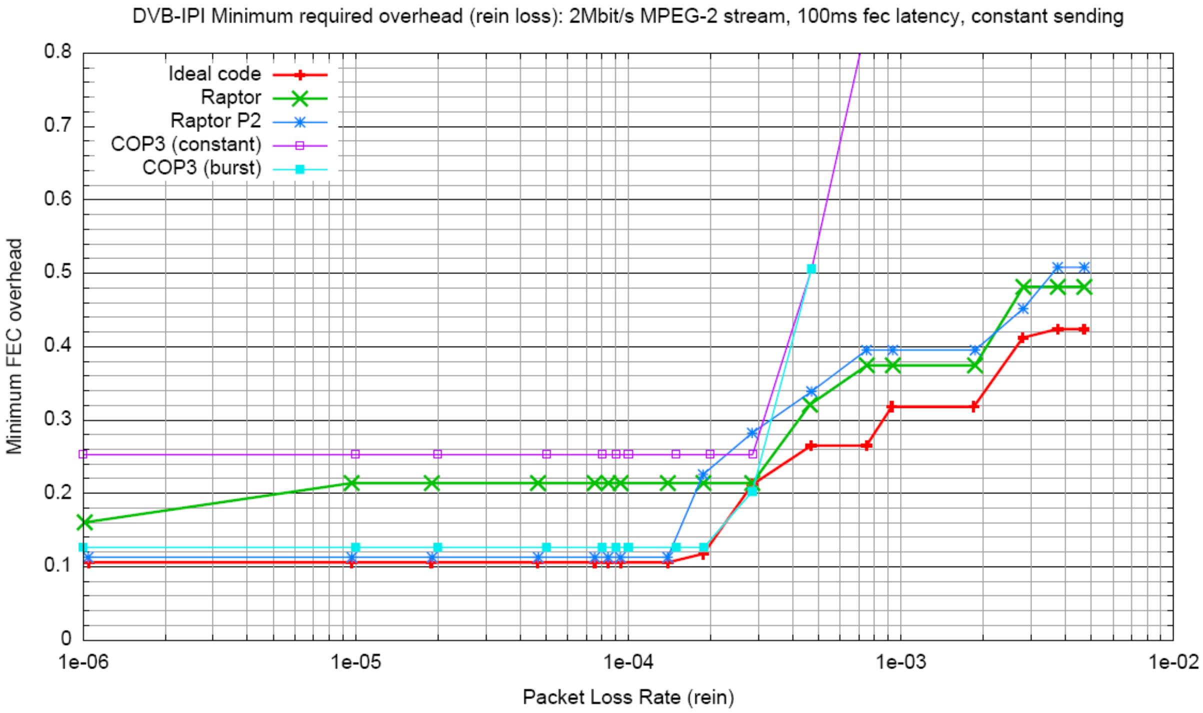

Figure A.9-7: 2 Mbit/s MPEG-2 Transport Stream, 100 ms latency, REIN Loss, constant sending

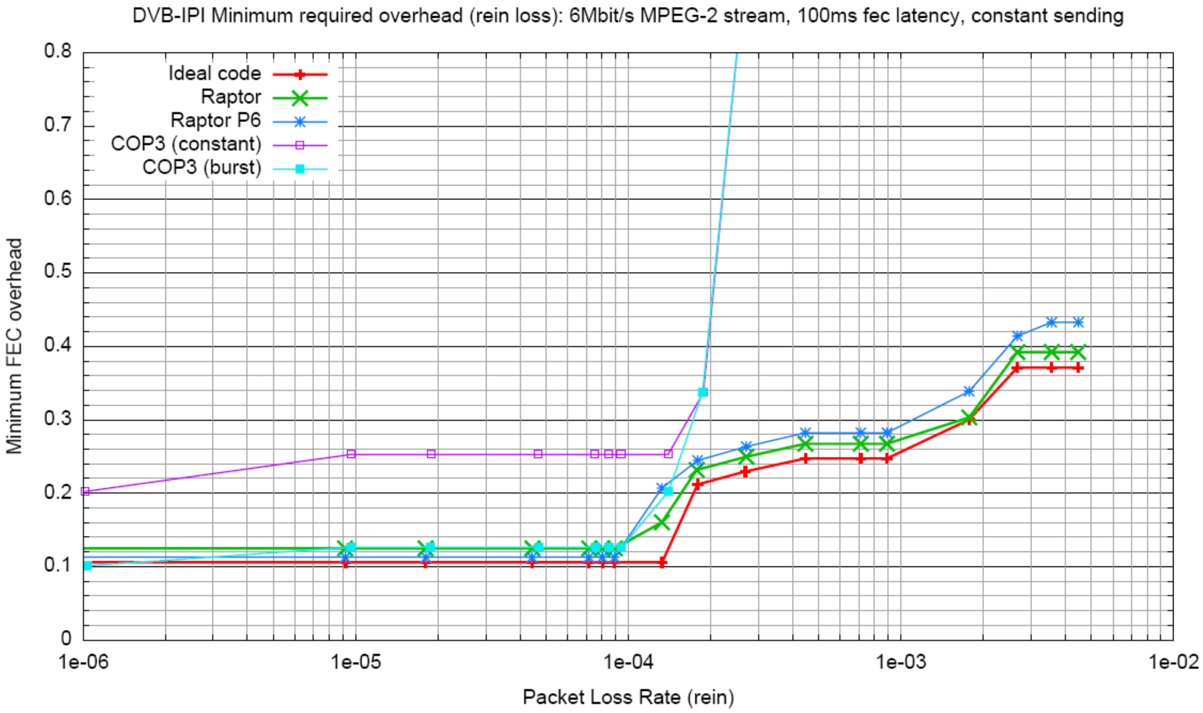

Figure A.9-8: 6 Mbit/s MPEG-2 Transport Stream, 100 ms latency, REIN Loss, constant sending

History

Document history		
V1.1.1	November 2006	Publication as TR 102 542
V1.2.1	April 2008	Publication