File Delivery over DVB-H, FLUTE

Filecasting

- **Filecasting == file broadcasting**
  - Non-real-time broadcasting of audio, video, and data files that are to be stored at the receiver for playback at a later time

- **File Delivery over Unidirectional Transport (FLUTE) protocol**
  - IETF RMT WG
  - RFC 3926 [1]

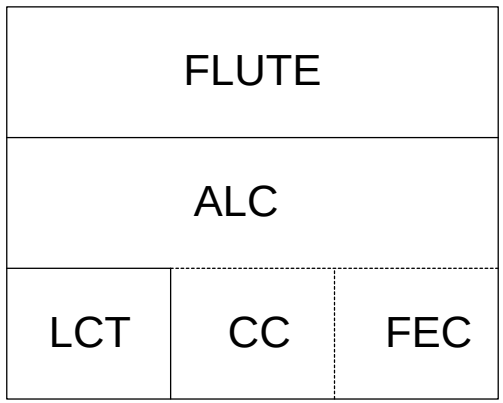
  “FLUTE is a protocol for the unidirectional delivery of files over the Internet, which is particularly suited to multicast networks. The specification builds on Asynchronous Layered Coding (ALC), the base protocol designed for massively scalable multicast distribution.”

FLUTE Basics

- **To unidirectional delivery of files**
  - Files are delivered as transport objects, with optional content encoding (e.g. gzip)
- **Distribution to large number of receivers**
  - No feedback from them
- **Supports IPv4 and IPv6**
  - No IP version specific parts in the FLUTE header
- **Can be used with both multicast and unicast User Datagram Protocol (UDP) delivery**
  - Particularly suited to multicast networks
- **Supports Any Source Multicast (ASM) and Source Specific Multicast (SSM) models**
- **Reliability through retransmissions and Forward Error Correction**

FLUTE Building Blocks

- **FLUTE builds on Asynchronous Layered Coding (ALC) Protocol Instantiation [2] of the Layered Coding Transport (LCT) Building Block [3]**
- **ALC combines the LCT Building Block, a Congestion Control (CC) Building Block and a Forward Error Correction (FEC) Building Block [4] to provide congestion controlled reliable asynchronous delivery**
- **LCT provides transport level support for reliable content delivery and stream delivery protocols**
- **The use of CC and FEC building blocks with FLUTE is optional**



Forward Error Correction

- **Parity data to recover packet losses**
- **The use of Forward Error Correction defined in RFC 3453 [5]**
- **With FLUTE the default FEC code is Compact No-Code FEC [6]**
  - No actual FEC encoding or decoding
  - File segmentation into source blocks
  - FLUTE's algorithm for computing source block structure
    - Computes a source block structure so that all source blocks are as close to being equal length as possible
    - First number of source blocks share same larger length
    - Remaining second number of source blocks share the same smaller length
  - Encoding symbols contains only the source symbols

Forward Error Correction

- **Others**
  - XOR FEC [7]
  - Reed-Solomon FEC [7]
  - Parity Check Matrix-based FEC [7]
    - Low Density Parity Check (LDPC) FEC
    - Low Density Generator Matrix (LDGM) FEC
      - LDGM-Staircase
      - LDGM-Triangle
  - Raptor FEC [10]
    - Digital Fountain owns IPR

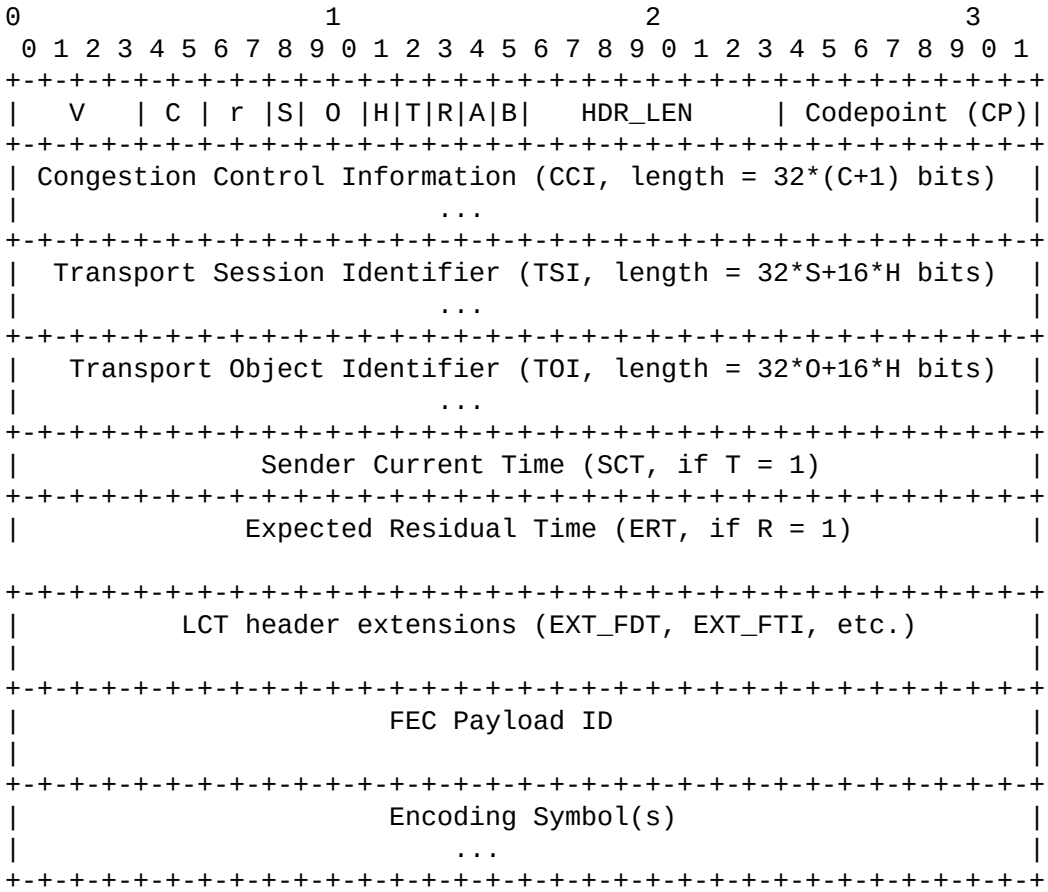
Congestion Control

- **Congestion Control (CC) building block(s) needed to enable co-existence of FLUTE and TCP traffic on the Internet**

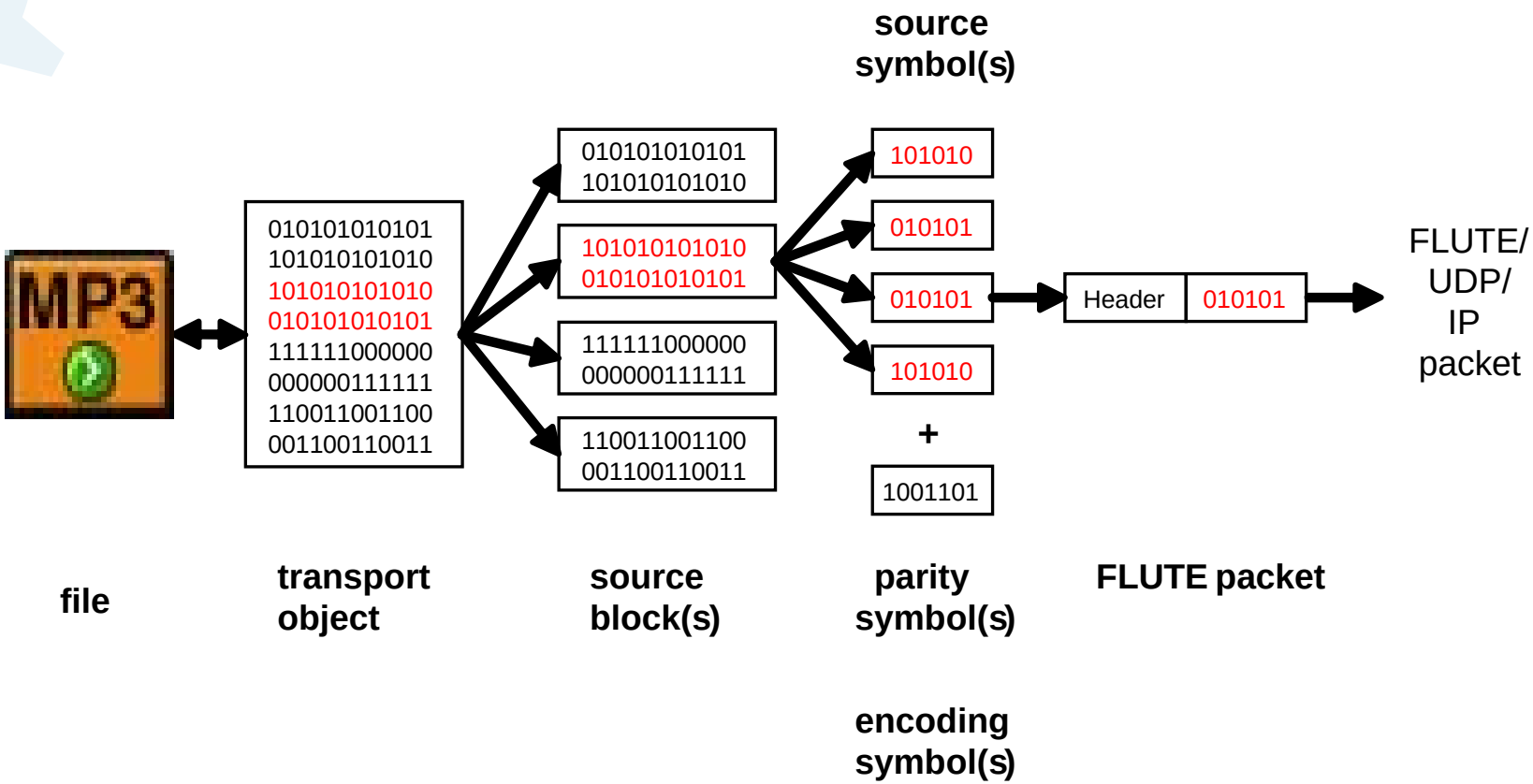
  "FLUTE is applicable for both Internet use, with a suitable congestion control building block, and provisioned/controlled systems, such as delivery over wireless broadcast radio systems."

- **Possible CC BB from IETF RMT WG**
  - Wave and Equation Based Rate Control (WEBRC) Building Block
    - RFC3738 (http://www.ietf.org/rfc/rfc3738.txt)
    - Receiver-driven
      - No feedback from receiver to sender

FLUTE Packet



Building up a FLUTE Packet



FLUTE Session

- **A FLUTE session (i.e. an ALC/LCT session) consists of one or more ALC/LCT channels**
  - Defined by the combination of a sender's IP address and an address associated with the channel by the sender
  - Congestion control and multiple rate delivery by using multiple channels (layers)
- **A receiver joins a channel to start receiving the data packets sent to the channel by the sender**
- **The receiver leaves the channel to stop receiving data packets from the channel**
- **SDP Descriptors for FLUTE I-D [8] describes the use of SDP to describe the parameters required to begin, join, receive data from, and end FLUTE sessions**
- **File Delivery Table (FDT) Instances together with FLUTE header fields gives the necessary parameters to identify, locate and restore the files at the receiver**
- **Sessions may be started without complete knowledge of their content**
  - Requires FDT Instances to gradually define file parameters during the session

SDP Descriptors for FLUTE

- **Example:**

```
v=0 o=user123 2890844526 2890842807 IN IP6 2201:056D::112E:144A:1E24
s=File delivery session example
i=More information
t=2873397496 2873404696
a=source-filter: incl IN IP6 * 2001:210:1:2:240:96FF:FE25:8EC9
a=flute-tsi:1
a=flute-ch:2
a=FEC-declaration:0 encoding-id=0
a=FEC-declaration:1 encoding-id=128; instance-id=0
a=content-desc:http://www.example.com/flute-sessions/session001
m=application 12345 FLUTE/UDP 0
c=IN IP6 FF1E:03AD::7F2E:172A:1E24
a=FEC:0
m=application 12346 FLUTE/UDP 0
c=IN IP6 FF1E:03AD::7F2E:172A:1E25
a=FEC:1
```

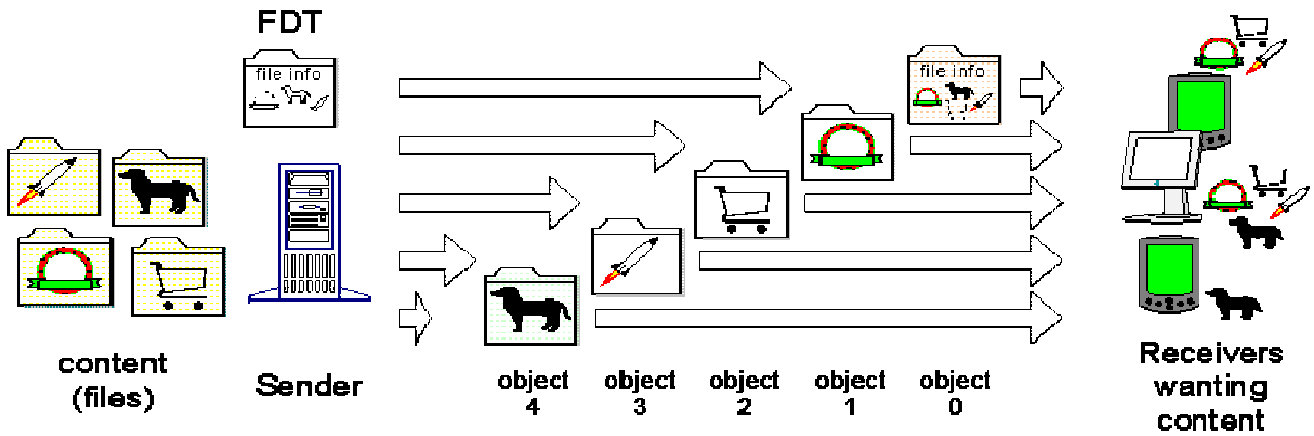
File Delivery Table Instance

- **One FDT Instance can describe all or part of the files for the FLUTE session**
- **An example FDT Instance:**

```
<?xml version="1.0" encoding="UTF-8"?>
<FDT-Instance xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xmlns:fl="http://www.example.com/flute"
              xsi:schemaLocation="http://www.example.com/flute-fdt.xsd"
              Expires="2890842807">
 <File Content-Location="www.example.com/menu/tracklist.html"
       TOI="1"
       Content-Type="text/html"/>
 <File Content-Location="www.example.com/tracks/track1.mp3"
       TOI="2"
       Content-Length="6100"
       Content-Type="audio/mp3"
       Content-MD5="Eth76GlkJU45sghK"
       Some-Private-Extension-Tag="abc123"/>
</FDT-Instance>
```

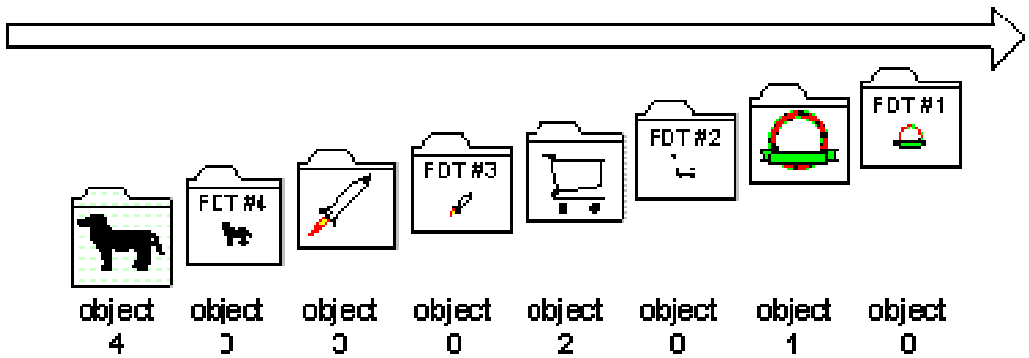
FLUTE Session Example 1

Four files described by one FDT Instance, and that FDT Instance is delivered before the files (Figure used with permission from Rod Walsh, published in "Advances in Mass Media Delivery to Mobiles", MIPS2004 Tutorial, http://mips2004.imag.fr/tutorials.php#tutorial2)



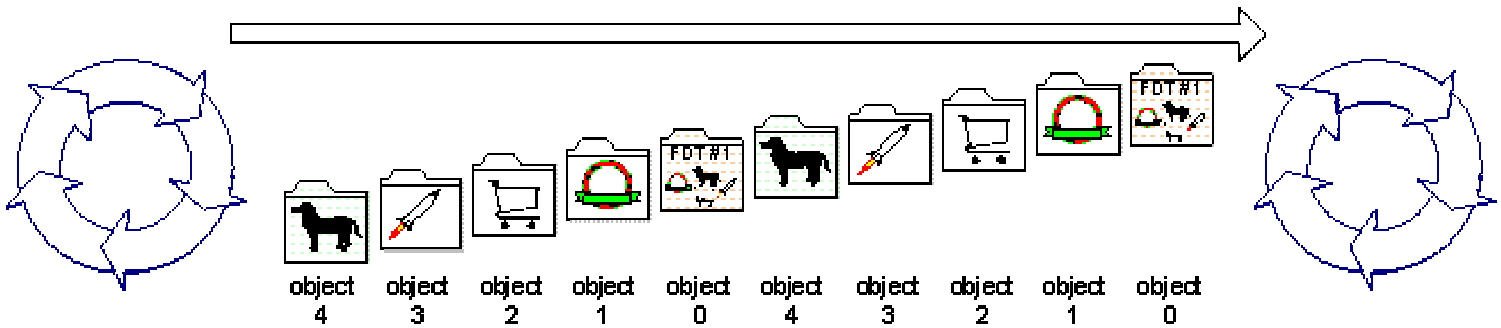
FLUTE Session Example 2

Four files described by four FDT Instances, and each FDT Instance is delivered before corresponding file (Figure used with permission from Rod Walsh, published in "Advances in Mass Media Delivery to Mobiles", MIPS2004 Tutorial, http://mips2004.imag.fr/tutorials.php#tutorial2)



FLUTE Session Example 3

Retransmission of four files described by one FDT Instance, and that FDT Instance is delivered before the files (Figure used with permission from Rod Walsh, published in "Advances in Mass Media Delivery to Mobiles", MIPS2004 Tutorial, http://mips2004.imag.fr/tutorials.php#tutorial2)



Existing FLUTE Implementations

- **TUT (Tampere University of Technology)**
  - Open source
  - http://www.atm.tut.fi/mad
  - jani.peltotalo@tut.fi
- **INRIA (Institut National de Recherche en Informatique et en Automatique)**
  - Open source
  - http://www.inrialpes.fr/planete/people/roca/mcl/mcl.html
  - vincent.roca@inrialpes.fr
- **University of Bremen**
  - Open source
  - ftp://ftp.informatik.uni-bremen.de/home/logic/flute/
  - jo@tzi.uni-bremen.de
- **Nokia**
  - Proprietary
  - rod.walsh@nokia.com
- **Digital Fountain**
  - Proprietary, productized
  - luby@digitalfountain.com

MAD-FLUTE

- **Implemented using C-language**
- **Linux and Windows OS**
  - Ported also to FreeBSD by Janos Mohacsi
    - http://www.freshports.org/net/mad_fcl/
    - janos.mohacsi@bsd.hu
- **IPv4 and IPv6 capable**
- **Unicast and multicast UDP**
- **Supports ASM and SSM**
- **FEC**
  - Compact No-Code FEC
  - XOR FEC
  - Reed-Solomon FEC
- **CC**
  - No CC
  - Receiver-driven Layered Congestion Control [11]
- **Used in 6NET tests**
  - Stig Venaas is transmitting new I-Ds using IPv6 SSM sender
  - http://www.m6bone.net/article.php3?id_article=88
- **Point-to-point file repair prosedure under work**

FLUTE Interoperability

- **TUT, INRIA, University of Bremen and Nokia implementations tested against each other in one-to-one, one-to-many, many-to-many cases**
- **FLUTE Interoperability Testing Guidelines [9] I-D written as a result of these test**
  - To illustrate common implementation errors
  - To help new implementers to test their implementations against existing ones

FLUTE in DVB

- **IP Datacasting over DVB-H (DVB-H / IPDC)**
  - FLUTE recommended transport protocol for announcement files in Service Discovery Channel [12]
  - FLUTE proposed transport protocol for push file delivery [12]

FLUTE in 3GPP

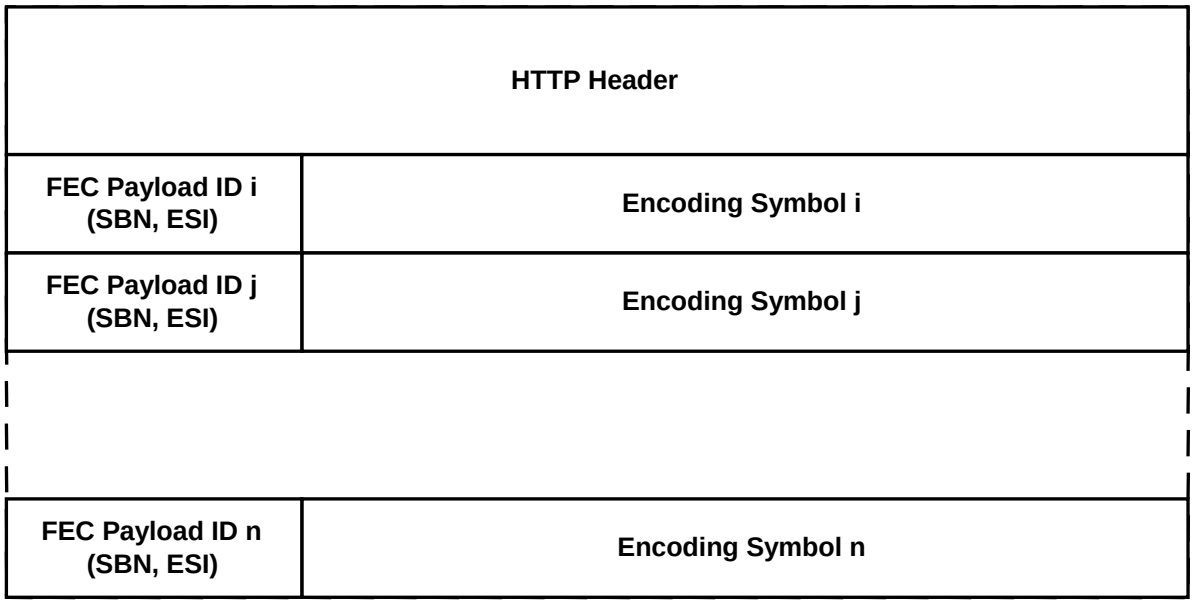
- **3GPP Multimedia Broadcast Multicast Service (MBMS)**
  - FLUTE selected for MBMS download delivery method [13]
  - Additional file repair procedure (requires return channel)
    - HTTP file repair request
    - Point-to-point repair
      - The file repair response message consists of HTTP header and file repair response payload (HTTP payload)
    - Point-to-multipoint repair
      - The file repair response message consists of HTTP header
        - » informs that point-to-multipoint repair is used instead of point-to-point repair
      - Actual file repair response payload using broadcast/multicast

File Repair

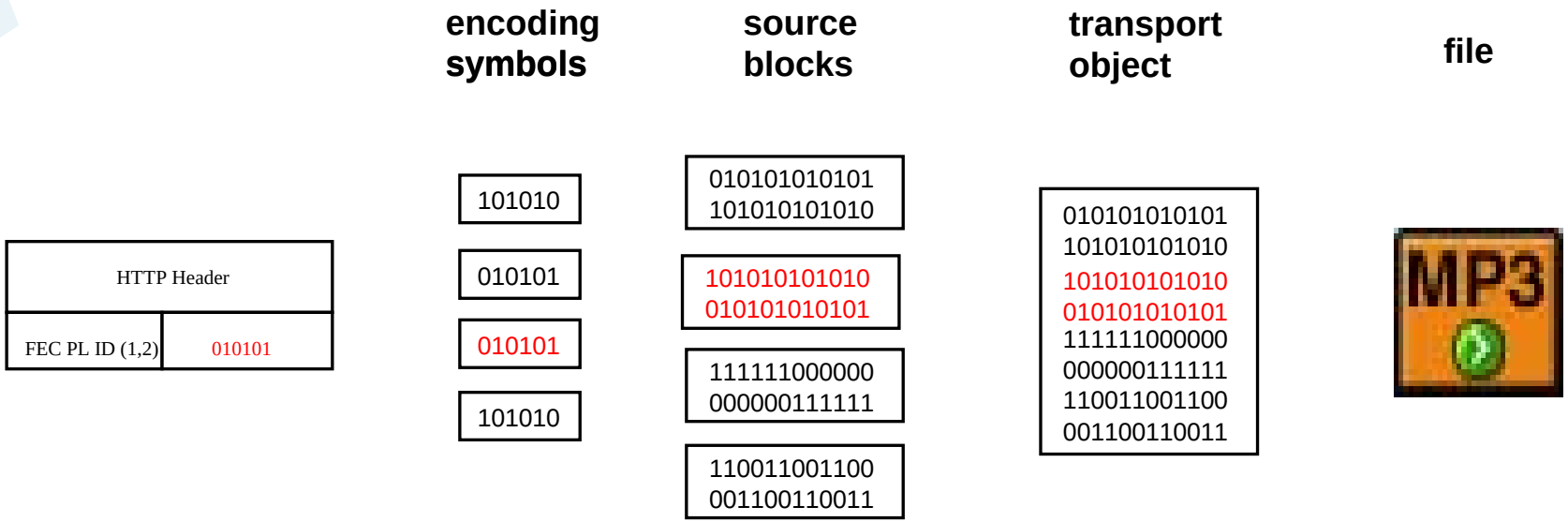
- **Example HTTP GET request:**

GET /cgi-bin/*mbms-rel6-FLUTE-repair.cgi?fileURI=file://www.example.com/dir/example.txt&SBN=5;ESI=12&SBN=20;ESI=27* ***HTTP/1.1***

- **File Repair Response Message Format:**



File Repair Example



References

[1] Paila, T., Luby, M., Lehtonen, R., Roca, V. and R. Walsh: FLUTE - File Delivery over Unidirectional Transport. RFC 3926, October 2004

[2] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L. and J. Crowcroft: Asynchronous Layered Coding (ALC) Protocol Instantiation. RFC 3450, December 2002

[3] Luby, M., Gemmell, J., Vicisano, L., Rizzo, L., Handley, M. and J. Crowcroft: Layered Coding Transport (LCT) Building Block. RFC 3451, December 2002

[4] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M. and J. Crowcroft: Forward Error Correction (FEC) Building Block. RFC 3452, December 2002

[5] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M. and J. Crowcroft: The Use of Forward Error Correction (FEC) in Reliable Multicast. RFC 3453, December 2002

[6] Luby, M. and L. Vicisano: Compact Forward Error Correction (FEC) Schemes. RFC 3695, February 2004

[7] Peltotalo, J., Peltotalo, S. and V. Roca: Simple XOR, Reed-Solomon, and Parity Check Matrix-based FEC Schemes. IETF, draft-peltotalo-rmt-bb-fec-supp-xor-pcm-rs-00.txt (Expired), June 2004

[8] Mehta, H., Walsh, R., Curcio, I., Peltotalo, J. and S. Peltotalo: SDP Descriptors for FLUTE. IETF, draft-mehta-rmt-flute-sdp-02.txt (Work in Progress), November 2004

[9] Mehta, H., Walsh, R., Roca, V., Peltotalo, J. and S. Peltotalo: FLUTE Interoperability Testing Guidelines. IETF, draft-mehta-rmt-flute-iop-03.txt (Work in Progress), December 2004

[10] Luby, M. and A. Shokrollahi: Raptor Forward Error Correction (FEC) Schemes for Objects. IETF, draft-luby-rmt-bb-fec-raptor-object-00.txt (Work in Progress), October 2004

References

[11] Rizzo, L., Vicisano, L. and J. Crowcroft: The RLC multicast congestion control algorithm. Submitted to IEEE Network - special issue multicast

[12] DVB, “IP Datacast Baseline Specification: Specification of Interface I_MT”, DVB A080, April 2004

[13] 3GPP TS 26.346 V2.0.0 (2005-03), 3rd Generation Partnership Project;Technical Specification Group Services and System Aspects;Multimedia Broadcast/Multicast Service (MBMS);Protocols and codecs(Release 6)