Pseudo Content Delivery Protocol (CDP) for Protecting Multiple Source Flows in FEC Framework
draft-ietf-fecframe-pseudo-cdp-01

Status of this Memo

   This Internet-Draft is submitted to IETF in full conformance with the
   provisions of BCP 78 and BCP 79. This document may contain material
   from IETF Documents or IETF Contributions published or made publicly
   available before November 10, 2008. The person(s) controlling the
   copyright in some of this material may not have granted the IETF
   Trust the right to allow modifications of such material outside the
   IETF Standards Process. Without obtaining an adequate license from
   the person(s) controlling the copyright in such materials, this
   document may not be modified outside the IETF Standards Process, and
   derivative works of it may not be created outside the IETF Standards
   Process, except to format it for publication as an RFC or to
   translate it into languages other than English.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as Internet-
   Drafts.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   The list of current Internet-Drafts can be accessed at
   http://www.ietf.org/ietf/1id-abstracts.txt.

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html.

   This Internet-Draft will expire on September 8, 2009.

Copyright Notice

Abstract

   This document provides a pseudo Content Delivery Protocol (CDP) to
   protect multiple source flows with one or more repair flows based on
   the FEC Framework document and the Session Description Protocol (SDP)
   elements defined for the framework. The purpose of the document is
   not to provide a full-pledged protocol, but to show how the defined
   framework and SDP elements can be combined together to design a CDP.

Table of Contents

1. **Introduction**

   The Forward Error Correction (FEC) Framework (described in
   [I-D.ietf-fecframe-framework]) and SDP Elements for FEC Framework
   (described in [I-D.ietf-fecframe-sdp-elements]) together define
   mechanisms sufficient enough to build an actual Content Delivery
   Protocol (CDP) with FEC protection. Methods to convey FEC Framework
   Configuration Information (described in
   [I-D.ietf-fecframe-config-signaling]) on the other hand provides the
   signaling protocols that may be used as part of CDP to communicate
   FEC Scheme-Specific Information from FEC sender to a single as well
   as multiple FEC receivers. This document aims at providing a
   guideline on how the mechanisms defined in
   [I-D.ietf-fecframe-framework] and [I-D.ietf-fecframe-sdp-elements]
   can be sufficiently used to design a CDP over a non-trivial scenario,
   namely protection of multiple source flows with one or more repair
   flows.

   In particular, we provide clarifications and descriptions on how:

   o source and repair flows may be uniquely identified,

   o source blocks may be generated from one or more source flows,

   o repair flows may be paired with the source flows,

   o the receiver explicitly and implicitly identifies individual
      flows,

   o source blocks are regenerated at the receiver and the missing
      source symbols in a source block are recovered.

2. **Requirements Notation**

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in [RFC2119].

3. **Definitions/Abbreviations**

   This document uses the following definitions. For further
   definitions that apply to FEC Framework in general, see
   [I-D.ietf-fecframe-framework].

CDP: Content Delivery Protocol.

FEC: Forward Error Correction.

Source Flow: The packet flow or flows to which FEC protection is to
   be applied.

Repair Flow: The packet flow or flows carrying FEC data.

Transport Protocol: The protocol used for transport of the source
   data flow being protected.

FEC Scheme: A specification which defines the additional protocol
   aspects required to use a particular FEC code with the FEC
   framework.

Source Block: The group of source data packets which are to be FEC
   protected as a single block.

Source FEC Payload ID: An FEC Payload ID specifically for use with
   source packets.

Repair FEC Payload ID: An FEC Payload ID specifically for use with
   repair packets.

[4]. Construction of a Repair Flow from Multiple Source Flows

At the sender side, CDP constructs the source blocks (SB) by
multiplexing transport payloads from multiple flows (See Figure 1 and
Figure 2). According to the FEC Framework, each source block is FEC
protected separately. Each source block is given to the specific FEC
encoder used within the CDP as input and as the outputs Explicit
Source FEC Payload ID, Repair FEC Payload ID, and Repair Payloads
corresponding to that source block are generated. Note that Explicit
Source FEC payload ID is optional and if CDP has implicit means of
constructing the source block at the sender/receiver (e.g., by using
any existing sequence numbers in the payload), the Explicit Source
FEC payload ID might not be output.

```
               +------------+
   s_1 -------> |            |
     .  Source  | Source     |     +--------+ +--------+ +--------+
     .  Flows   | Block      |=> ..|SB_(j+1)| |  SB_j  | |SB_(j-1)| ..
   s_n -------> | Generation |     +--------+ +--------+ +--------+
               +------------+
```

                  Figure 1: Source Block generation for an FEC scheme

    Figure 2 shows the structure of a source block. A CDP MUST clearly
    specify which payload corresponds to which source flow and the length
    of each payload.

        <------------------ Source Block (SB) ------------------->

        +-------.,.-----+-------.,.-----+-   -+-------.,.-----+
        | Payload_1 | Payload_2 | . . .| Payload_n |
        +-------.,.-----+-------.,.-----+-   -+-------.,.-----+
        _____ _____|_____ _____| |_____ _____|
              \/   \/   \/
         FID_1,Len_1   FID_2,Len_2   FID_n,Len_n

                   Figure 2: Structure of a Source Block

    Flow ID (FID) value provides a unique short-hand identifier for the
    source flows. FID is specified and associated with the possibly
    wildcarded tuple of {Source IP Address, Destination IP Address,
    Source Transport Port, Destination Transport Port, Transport
    Protocol} in the SDP file. When wildcarded, certain fields in the
    tuple are not needed for distinguishing the source flows. The tuple
    is carried in the IP and transport headers of the source packets.
    Since FID is utilized by the CDP and FEC scheme to distinguish
    between the source packets, the tuple MUST have a one-to-one mapping
    to a valid FID. This point will be clearer in the specific example
    given later in this section. The length of FID must be a priori
    fixed and known to both the receiver and sender. Alternatively, it
    might be specified in the FEC-Scheme-Specific Information field in
    the SDP element [I-D.ietf-fecframe-sdp-elements].

    The payload length (Len) information is needed to figure out how many
    bits, bytes, or symbols (depending on the FEC scheme) from a
    particular source flow are included in the source block. If the
    payload is not an integer multiple of the specified symbol length,
    the remaining portion is padded with zeros (See Figure 3 and
    Figure 4).

                                              +------+
        +--------+ +--------+ +--------+ | | -------> r_1
     .. |SB_(j+1)| | SB_j | |SB_(j-1)| .. --> | FEC | Repair   .
        +--------+ +--------+ +--------+ |Scheme| Flows   .
                                              | | -------> r_k
                                              +------+
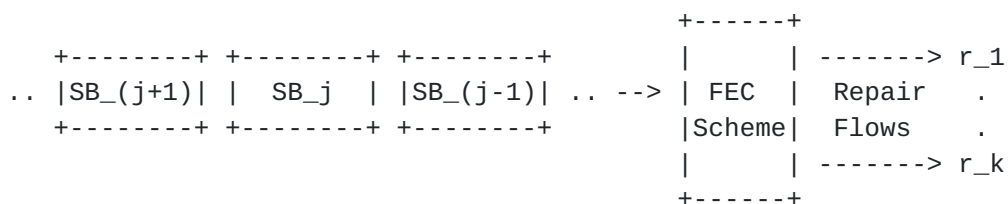
Figure 3: Repair flow generation by an FEC scheme

```
     <----------------- Source Block (SB) ------------------->
     |         |          |            |          |          |
     +-------.,.-----+-------.,.-----+-     -+-------.,.-----+ |
     |  Payload_1   |   Payload_2   | . . . |   Payload_n   |0|
     +-------.,.-----+-------.,.-----+-     -+-------.,.-----+ |
     |         |          |            |          |          |
     | Symbol_1 | Symbol_2 | Symbol_3 |        . . .    | Symbol_m |
     |<-------->|<-------->|<-------->|                |<-------->|


                         +------+
       Symbol_1,..,Symbol_m => |  FEC  | => Symbol_u,..,Symbol_1
                         | Enc. |
                         +------+
```
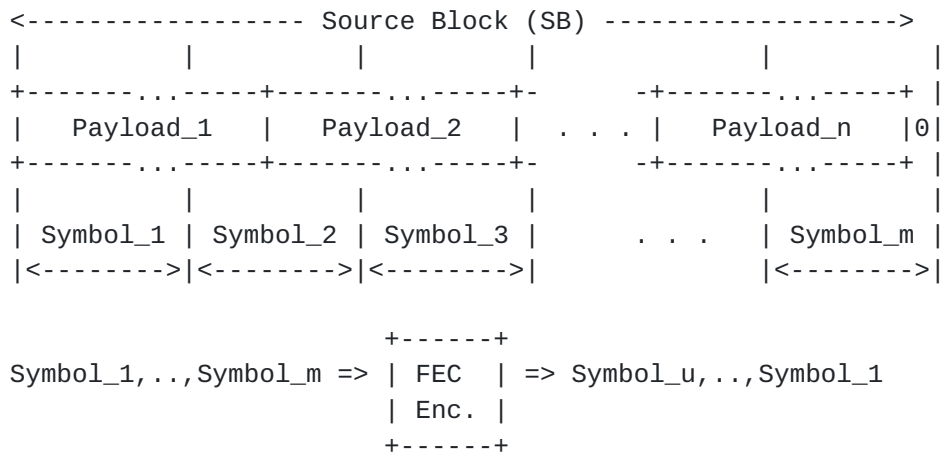
Figure 4: Repair flow payload generation

FEC schemes typically expect a source block of certain size, say m
symbols. Therefore, the FEC encoder divides each source block into m
symbols (with some padding if the source block is shorter than the
expected m symbols) and generates u repair symbols which are
functions of the m symbols in the original source block. The repair
symbols are grouped by the FEC scheme into repair payloads with each
repair payload assigned a Repair FEC Payload ID in order to associate
each repair payload with a particular source block at the receiver.
If the payloads in a given source block have sequence numbers that
can uniquely specify their location in the source block, an Explicit
Source FEC Payload ID may not be generated for these payloads.
Otherwise, Explicit Source FEC Payload IDs are generated for each
payload and indicate the order the payloads appear in the source
block.

Note that FID and length information are not actually transmitted
with the source payloads since both information can be gathered by
other means as it will be clear in the next sections.

4.1. Example: Two Source Flows Protected by a Single Repair Flow

In this section, we present an example of source flow and repair flow
generation by the CDP. We have two source flows with flow IDs of 0
and 1 to be protected by a single repair flow (See Figure 5). The
first source flow is multicast to 224.1.1.1 and the second source
flow is multicast to 224.1.1.2. Both flows use the port number
30000. The SDP description below states that the source flow defined
by the tuple {*,224.1.1.1,*,30000} is identified with FID=0 and the
source flow defined by the tuple {*,224.1.1.2,*,30000} is identified

with FID=1. The SDP description also states that the repair flow is
to be received at the multicast address of 224.1.2.1 and at port
30000.

```
                SOURCE FLOWS                | INSTANCE #1
              0: Source Flow |_____| 2: Repair Flow
              1: Source Flow |
```

   Figure 5: Example: Two source flows and one repair flow

```
v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=FEC Framework Examples
t=0 0
a=group:FEC S1 S2 R1
m=video 30000 RTP/AVP 100
c=IN IP4 224.1.1.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S1
m=video 30000 RTP/AVP 101
c=IN IP4 224.1.1.2/127
a=rtpmap:101 MP2T/90000
a=fec-source-flow: id=1
a=mid:S2
m=application 30000 udp/fec
c=IN IP4 224.1.2.1/127
a=fec-repair-flow: encoding-id=0; ss-fssi=5hu=
a=repair-window: 200
a=mid:R1
```

Figure 6 shows the first and the second source blocks (SB_1 and SB_2)
generated from these two source flows. In this example, SB_1 is of
length 10000 bytes. Suppose that the FEC scheme uses a symbol length
of 512 bytes. Then SB_1 can be divided into 20 symbols after padding
the source block for 240 bytes. Assume that the FEC scheme is
rate-2/3 erasure code, hence, it generates 10 repair symbols from 20
original symbols for SB_1. On the other hand, SB_2 is 7000-byte long
and can be divided into 14 symbols after padding 168 bytes. Using
the same encoder, suppose that 7 repair symbols are generated for
SB_2.

```
               <-------- Source Block 1 -------->
               +------------+------------------+
               | $1 $2 $3 $4| #1 #2 #3 #4 #5 #6 | 0..00
               +------------+------------------+
               _____  _____/
                                \/
                   @1 @2 @3 @4 @5 @6 @7 @8 @9 @10


               <---- Source Block 2 ---->
               +----------------+-------+
               | $5 $6 $7 $8 $9 | #7 #8 |0..00
               +----------------+-------+
               _____  _____/
                               \/
                 @11 @12 @13 @14 @15 @16 @17

               $: 1000-byte payload from source flow 1
               #: 1000-byte payload from source flow 2
               @: Repair symbol
```

               Figure 6: Source block with two source flows

   The information on the unit of payload length, FEC scheme, symbol
   size, and coding rates can be specified in the FEC Scheme-Specific
   Information (FSSI) field of the SDP element. If the values of the
   payload lengths from each source flow and the order of appearance of
   source flows in every source block are fixed during the session,
   these values may be also provided in the FSSI field. To carry FSSI
   information to the FEC receivers, one may utilize the signaling
   methods described in [I-D.ietf-fecframe-config-signaling]. In our
   example, we will consider the case where the ordering is fixed and
   known both at the sender and the receiver, but the payload lengths
   will be variable from one source block to another. We assume that
   the payload of a source flow with an FID smaller than another flow's
   FID precedes other payloads in a source block.

   The FEC scheme gets the source blocks as input and generates the
   parity blocks for each source block to protect the whole source
   block. In the example, the repair payloads for SB_1 consist of 512-
   byte symbols, denoted by @1 to @10. Similarly @11 to @17 constitute
   the repair payloads for SB_2. The FEC scheme outputs the repair
   payloads along with the Repair FEC Payload IDs. In our example,
   Repair FEC Payload ID provides information on the source block
   sequence number and the order the repair symbols are generated. For
   instance @3 is the third FEC repair symbol for SB_1 and the three
   tuple {@3, SB_1,3} can uniquely deliver this information. In our
   example, the FEC scheme also provides Explicit Source FEC Payload IDs

that carry information to indicate which source symbols correspond to which source block sequence number and its relative position in the source block. For instance the two tuple {SB_2,2} can be attached to $6 as the Explicit Source FEC Payload ID to indicate that $6 is protected together with packets belonging to SB_2, and $6 is the second payload in SB_2.

The source packets are generated from the source symbols by concatenating consecutive symbols in one packet. There SHOULD NOT be any fragmentation of a source symbol, e.g., symbols #7 and #8 can be concatenated in one transport payload of 2000-bytes (The implementation SHOULD make sure that the size of the resulting source packet - payload plus the overhead - is not larger than the path MTU), but one portion of symbol #7 SHOULD NOT be put in one source packet and the remaining portion in another source packet. The simplest implementation is to place each source symbol in a different source packet as shown in Figure 7.

```
          +------------------------------------+
          |        IP header {224.1.1.1}       |
          +------------------------------------+
          |       Transport header {30000}     |
          +------------------------------------+
          |   Original Transport Payload {$6}  |
          +------------------------------------+
          |   Source FEC Payload ID  {SB_2,2}  |
          +------------------------------------+
```

             Figure 7: Example of a source packet

The repair packets are generated from the repair symbols belonging to the same source block by grouping consecutive symbols in one packet. There should not be any fragmentation of a repair symbol, e.g., symbols @4, @5, and @6 can be concatenated in one transport payload of 1536-bytes, but @6 SHOULD NOT be divided into smaller sub-symbols and spread over multiple repair packets. The Repair FEC Payload ID MUST carry sufficient information for the decoding process and in our example indicating source block sequence number, length of each source payload, and the order that the first parity block in a repair packet is generated are sufficient. The exact header format of Repair FEC Payload ID may be specified in the FSSI field of the SDP element. In Figure 8 for instance, the repair symbols @4, @5, and @6 are concatenated together. The Payload ID {SB_1,4,4,6} states that the repair symbols protect SB_1, the first repair symbol in the payload is generated as the 4th symbol and the source block consists of two source flows carrying 4 and 6 packets from each.

```
+------------------------------------+
|          IP header {224.1.2.1}     |
+------------------------------------+
|        Transport header {30000}    |
+------------------------------------+
| Repair FEC Payload ID {SB_1,4,4,6} |
+------------------------------------+
|        Repair Symbols {@4,@5,@6}   |
+------------------------------------+
```

Figure 8: Example of a repair packet

5. Reconstruction of Source Flows from Repair Flow(s)

5.1. Example: Multiple Source Flows Protected by a Single Repair Flow

At the receiver, source flows 1 and 2 are received at
{224.1.1.1,30000} and {224.1.1.2,30000}, while the repair flow is
received at {224.1.2.1,30000}. The CDP can map these tuples to the
flow IDs using the SDP elements. Accordingly, the payloads received
at {224.1.1.1,30000} and {224.1.1.2,30000} are mapped to flow IDs 0
and 1, respectively.

The CDP passes the flow IDs and received payloads along with the
Explicit Source FEC Payload ID to the FEC scheme defined in the SDP
description. The CDP also passes the received repair packet payloads
and Repair FEC Payload ID to the FEC scheme. The FEC scheme can
construct the original source block with missing packets by using the
information given in the FEC Payload IDs. The FEC Repair Payload ID
provides the information that SB_1 has packets from two flows with 4
packets from the first one and 6 packets from the second one. Flow
IDs state that the packets from source flow 0 precedes the packets
from source flow 1. Explicit Source FEC Payload IDs on the other
hand provide the information about which source payload appears in
what order. Therefore, the FEC scheme can depict an source block
with exact locations of the missing packets. Figure 9 depicts the
case for SB_1. Since the original source block with missing packets
can be constructed at the decoder and the FEC scheme knows the coding
rate (e.g., it might be carried in the FSSI field in the SDP
description), a proper decoding operation can start as soon as the
repair symbols are provided to the FEC scheme.

```
              <-------- Source Block 1 -------->
              +------------+-------------------+
              | $1 $2 X  X | #1 X  #3 #4 #5 #6 |
              +------------+-------------------+

              O: Symbols received from the source flow 1 for SB_1
              #: Symbols received from the source flow 2 for SB_1
              X: Lost source symbols
```

                  Figure 9: Source block regeneration

   When the FEC scheme can recover any missing block while more repair
   symbols are arriving, it provides the recovered blocks along with the
   source flow IDs of the recovered blocks as outputs to the CDP. The
   receiver knows how long to wait to repair the remaining missing
   packets (e.g., specified by the 'repair-window' attribute in the SDP
   description). After the associated timer expires, the CDP hands over
   whatever could be recovered from the source flow to the application
   layer and continues with processing the next source block.

6. Security Considerations

   For the general security considerations related to the FEC Framework,
   refer to [I-D.ietf-fecframe-framework]. There are no additional
   security considerations that apply to this document.

7. IANA Considerations

   There are no IANA related issues considered in this document.

8. Acknowledgments

   The authors would like to thank the FEC Framework Design Team for
   their inputs, suggestions and contributions.

9. Normative References

   [I-D.ietf-fecframe-framework]
              Watson, M., "Forward Error Correction (FEC) Framework",
              draft-ietf-fecframe-framework-03 (work in progress),
              October 2008.

   [I-D.ietf-fecframe-sdp-elements]
              Begen, A., "SDP Elements for FEC Framework",

                    draft-ietf-fecframe-sdp-elements-02 (work in progress),
                    November 2008.

   [I-D.ietf-fecframe-config-signaling]
                    Asati, R., "Methods to convey FEC Framework Configuration
                    Information", draft-ietf-fecframe-config-signaling-01
                    (work in progress), November 2008.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
                    Requirement Levels", BCP 14, RFC 2119, March 1997.

Authors' Addresses

   Ulas C. Kozat
   DoCoMo USA Labs
   3240 Hillview Avenue
   Palo Alto, CA 94304-1201
   USA

   Phone: +1 650 496 4739
   Email: kozat@docomolabs-usa.com

   Ali Begen
   Cisco Systems
   170 West Tasman Drive
   San Jose, CA 95134
   USA

   Email: abegen@cisco.com