Internet Engineering Task Force (IETF)
Request for Comments: 6801
Category: Informational
ISSN: 2070-1721

U. Kozat
DOCOMO Innovations
A. Begen
Cisco

November 2012

Pseudo Content Delivery Protocol (CDP) for Protecting Multiple Source Flows in the Forward Error Correction (FEC) Framework

Abstract

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

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6801.

Table of Contents

1. Introduction

The Forward Error Correction (FEC) Framework (described in [RFC6363]) and SDP Elements for FEC Framework (described in [RFC6364]) 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 [RFC6695]), on the other hand, provide 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 provides a guideline on how the mechanisms defined in [RFC6363] and [RFC6364] 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:

- source and repair flows may be uniquely identified,
- source blocks may be generated from one or more source flows,
- repair flows may be paired with the source flows,
- the receiver explicitly and implicitly identifies individual flows, and
- source blocks are regenerated at the receiver and the missing source symbols in a source block are recovered.

2. Definitions/Abbreviations

This document uses all the definitions and abbreviations from Section 2 of [RFC6363] minus the RFC 2119 requirements language.

3. Construction of a Repair Flow from Multiple Source Flows

At the sender side, CDP constructs the source blocks (SBs) by multiplexing transport payloads from multiple flows (see Figures 1 and 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 the Explicit Source FEC Payload ID is optional, and if the CDP has an 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.

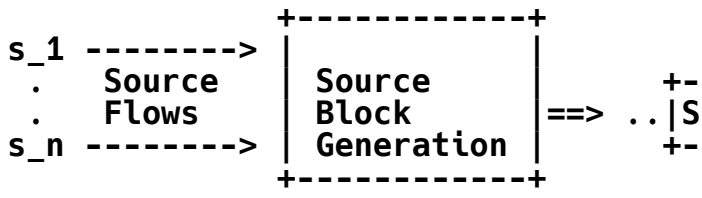

Figure 1: Source Block Generation for a FEC Scheme

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.

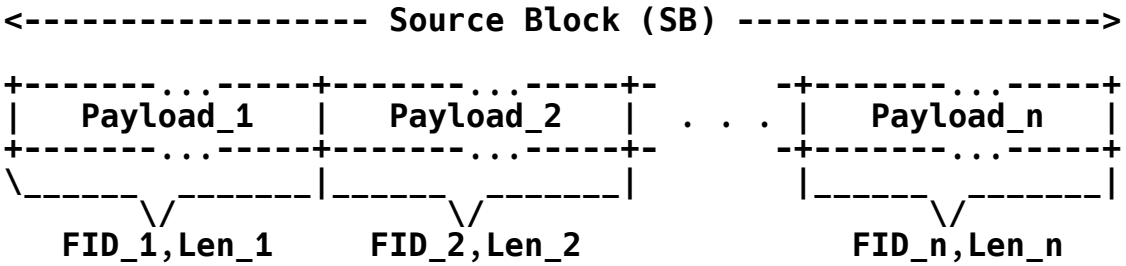

Figure 2: Structure of a Source Block

The Flow ID (FID) value provides a unique shorthand identifier for the source flows. FID is specified and associated with the possibly wildcarded tuple of {source IP address, source port, destination IP address, destination port, transport protocol} in the SDP description. 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 [RFC6364].

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 Figures 3 and 4).

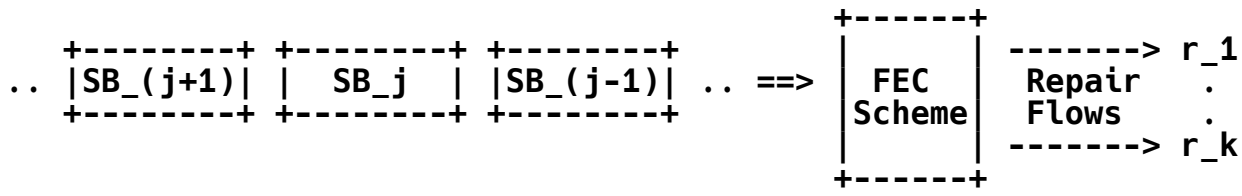

Figure 3: Repair Flow Generation by a FEC Scheme

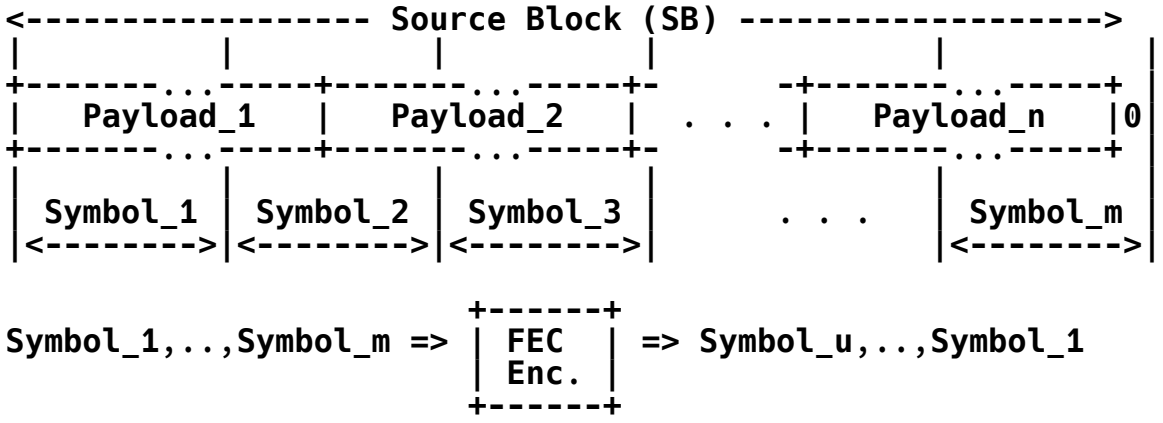

Figure 4: Repair Flow Payload Generation

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.

3.1. 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 233.252.0.1, and the second source flow is multicast to 233.252.0.2. Both flows use the port number 30000.

Figure 5: Example: Two Source Flows and One Repair Flow

The SDP description below states that the source flow defined by the tuple {*,*,233.252.0.1,30000} is identified with FID=0 and the source flow defined by the tuple {*,*,233.252.0.2,30000} is identified with FID=1 (via the 'id' parameter of the "fec-source-flow" attribute). The SDP description also states that the repair flow is to be received at the multicast address of 233.252.0.3 and at port 30000.

```
v=0
o=ali 1122334455 1122334466 IN IP4 fec.example.com
s=FEC Framework Examples
t=0 0
a=group:FEC-FR S1 S2 R3
m=video 30000 RTP/AVP 100
c=IN IP4 233.252.0.1/127
a=rtpmap:100 MP2T/90000
a=fec-source-flow: id=0
a=mid:S1
m=video 30000 RTP/AVP 101
c=IN IP4 233.252.0.2/127
a=rtpmap:101 MP2T/90000
a=fec-source-flow: id=1
a=mid:S2
m=application 30000 UDP/FEC
c=IN IP4 233.252.0.3/127
a=fec-repair-flow: encoding-id=0; ss-fssi=n:7,k:5
a=repair-window:150ms
a=mid:R3
```

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 bytes long and can be divided into 14 symbols after padding 168 bytes. Using the same encoder, suppose that seven repair symbols are generated for SB_2.

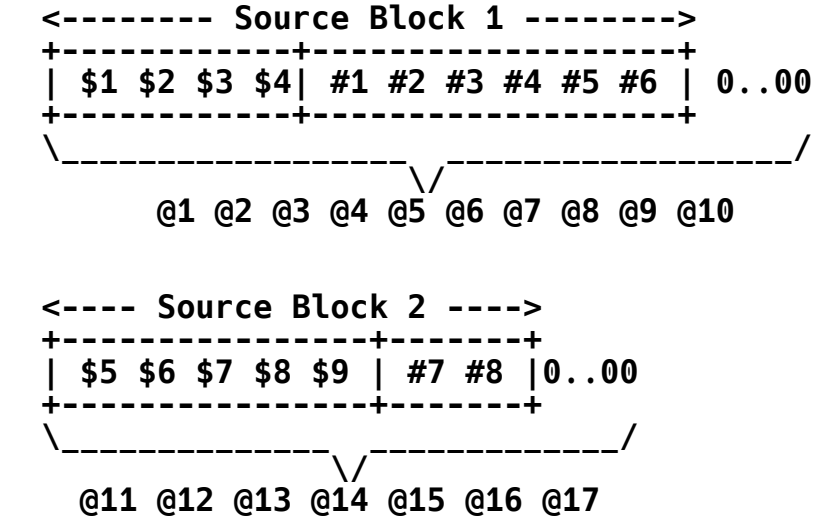

Figure 6: Source Block with Two Source Flows

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 use the signaling methods described in [RFC6695]. 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 constitutes 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 the 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.

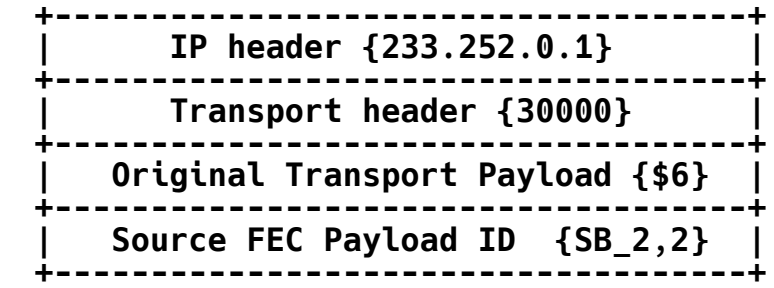

Figure 7: Example of a Source Packet for IPv4

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. In our example, for instance, indicating source block sequence number, length of each source payload, and the order that the first parity symbol in the repair packet among all the parity symbols generated

for the same source block is 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 fourth symbol and the source block consists of two source flows carrying four and six packets from each.

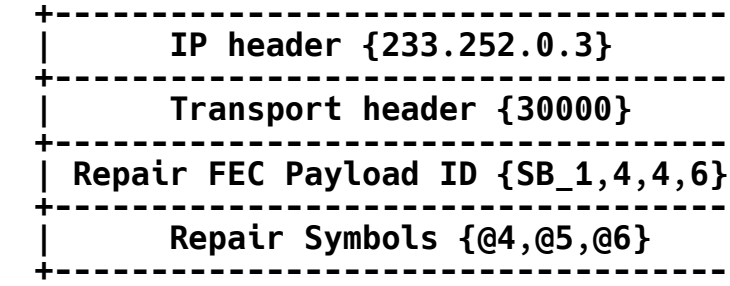

Figure 8: Example of a Repair Packet for IPv4

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

Here we provide an example for reconstructing multiple source flows from a single repair flow.

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

At the receiver, source flows 1 and 2 are received at {233.252.0.1,30000} and {233.252.0.2,30000}, while the repair flow is received at {233.252.0.3,30000}. The CDP can map these tuples to the flow IDs using the SDP elements. Accordingly, the payloads received at {233.252.0.1,30000} and {233.252.0.2,30000} 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 four packets from the first one and six packets from the second one. Flow IDs state that the packets from source flow 0 precede 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 a 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.

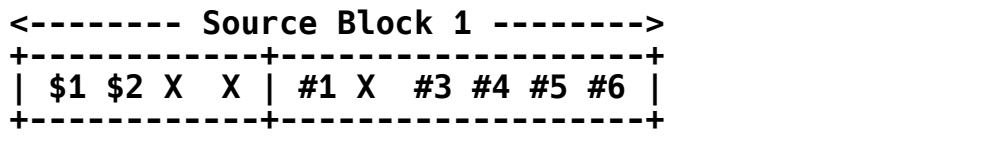

Figure 9: Source Block Regeneration

When the FEC scheme can recover any missing symbol 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.

5. Security Considerations

For the general security considerations related to the FEC Framework, refer to [RFC6363]. For the security considerations related to the SDP elements in the FEC Framework, refer to [RFC6364]. There are no additional security considerations that apply to this document.

6. Acknowledgments

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

7. Normative References

[RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, October 2011.

[RFC6364] Begen, A., "Session Description Protocol Elements for the Forward Error Correction (FEC) Framework", RFC 6364, October 2011.

[RFC6695] Asati, R., "Methods to Convey Forward Error Correction (FEC) Framework Configuration Information", RFC 6695, August 2012.

Authors' Addresses

Ulas C. Kozat
DOCOMO Innovations
3240 Hillview Avenue
Palo Alto, CA 94304-1201
USA

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

Ali Begen
Cisco
181 Bay Street
Toronto, ON M5J 2T3
Canada

EMail: abegen@cisco.com