FEC Framework Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 25, 2007

M. Watson
Digital Fountain

February 21, 2007

Forward Error Correction (FEC) Framework
draft-ietf-fecframe-framework-00

Status of this Memo

Copyright Notice

Abstract

This document describes for a framework for using forward error correction (FEC) codes with applications in the Internet to provide protection against packet loss. The framework supports applying Forward Error Correction to arbitrary packet flows and is primarily intended for streaming media. This framework can be used to define Content Delivery Protocols that provide Forward Error Correction for streaming media delivery or other packet flows. Content Delivery Protocols defined using this framework can support any FEC Scheme (and associated FEC codes) which is compliant with various requirements defined in this document. Thus, Content Delivery Protocols can be defined which are not specific to a particular FEC Scheme and FEC Schemes can be defined which are not specific to a particular Content Delivery Protocol.

Table of Contents

1. Introduction 4
2. Definitions/Abbreviations 5
3. Requirements notation 7
4. Architecture Overview 8
5. Procedural overview 10
5.1. General 10
5.2. Sender Operation 11
5.3. Receiver Operation 12
6. Protocol Specification 14
6.1. General 14
6.2. Structure of the source block 14
6.3. Packet format for FEC Source packets 14
6.4. Packet Format for FEC Repair packets 15
6.5. FEC Framework Configuration Information 16
6.6. FEC Scheme requirements 17
7. Transport Protocols 18
8. Session Description Protocol elements 19
8.1. udp/fec/<proto> transport protocol identifier 19
8.2. udp/fec transport protocol identifier 20
8.3. fec-declaration attribute 20
8.4. fec-oti-extension attribute 20
8.5. fec attribute 20
8.6. FEC media grouping semantics 20
8.7. SDP example 20
9. Congestion Control 21
9.1. Normative requirements 22
10. Security Considerations 24
11. IANA Considerations 25
12. Acknowledgments 26
13. References 27
Author's Address 28
Intellectual Property and Copyright Statements 29

1. Introduction

Many applications have a requirement to transport a continuous stream of packetised data from a source (sender) to one or more destinations (receivers) over networks which do not provide guaranteed packet delivery. Primary examples are media streaming applications such as broadcast, multicast or on-demand audio, video or multi-media.

Forward Error Correction is a well-known technique for improving reliability of packet transmission over networks which do not provide guaranteed packet delivery, especially in multicast and broadcast applications. The FEC Building Block defined in [4] provides a framework for definition of Content Delivery Protocols (CDPs) for object delivery (including, primarily, file delivery) which make use of separately defined FEC Schemes. Any CDP defined according to the requirements of the FEC Building Block can then easily be used with any FEC Scheme which is also defined according to the requirements of the FEC Building Block.

This document defines a framework for the definition of CDPs which provide for FEC protection of arbitrary packet flows over unreliable transports such as UDP. This document does not define a complete Content Delivery Protocol, but rather defines only those aspects that are expected to be common to all such Content Delivery Protocols.

This framework does not define how the flows to be protected are determined, nor how the details of the protected flows and the FEC streams which protect them are communicated from sender to receiver. It is expected that any complete Content Delivery Protocol specification which makes use of this framework will address these signalling requirements. However, this document does specify the information which is required by the FEC Framework at the sender and receiver - for example details of the flows to be FEC protected, the flow(s) that will carry the FEC protection data and an opaque container for FEC-Scheme-specific information. We also specify SDP [5] attributes which a Content Delivery Protocol MAY use to communicate this information.

FEC Schemes designed for use with this framework must fulfil a number of requirements defined in this document. Note that these requirements are different from those defined in [4] for FEC Schemes for object delivery. However there is a great deal of commonality and FEC Schemes defined for object delivery may be easily adapted for use with the framework defined here.

2. Definitions/Abbreviations

'FEC' Forward Erasure Correction.

'AL-FEC' Application Layer Forward Erasure Correction

'FEC Framework' A protocol framework for definition of Content Delivery Protocols using FEC, such as the framework defined in this document.

'Source data flow' The packet flow or flows to which FEC protection is to be applied.

'Repair data flow' The packet flow or flows carrying forward error correction data

'Source protocol' A protocol used for the source data flow being protected - e.g. RTP.

'Transport protocol' The protocol used for transport of the source data flow being protected - e.g. UDP, DCCP.

'Application protocol' Control protocols used to establish and control the source data flow being protected - e.g. RTSP.

'FEC Code' An algorithm for encoding data such that the encoded data flow is resiliant to data loss or corruption.

'FEC Scheme' A specification which defines the additional protocol aspects required to use a particular FEC code with the FEC framework, or (in the context of RMT), with the RMT FEC Building Block.

'Source Block' the group of source data packets which are to be FEC protected as a single block

'Protection amount' The relative increase in data sent due to the use of FEC.

FEC Framework Configuration Information: Information which controls the operation of the FEC Framework.

FEC Payload ID: Information which identifies the contents of a packet with respect to the FEC Scheme.

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.

Content Delivery Protocol (CDP): See [4].

3. 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 [1].

4. Architecture Overview

The FEC Framework is described in terms of an additional protocol layer between the transport layer (e.g. UDP or DCCP) and Application and Transport Protocols running over this transport layer. Examples of such protocols are RTP, RTCP, etc. As such, the data path interface between the FEC Framework and both underlying and overlying layers can be thought of as being the same as the standard interface to the transport layer - i.e. the data exchanged consists of datagram payloads each associated with a single transport flow identified by the standard 5-tuple { Source IP Address, Source Transport Port, Destination IP Address, Destination Transport Port, Transport Protocol }.

The FEC Framework makes use of an FEC Scheme, in a similar sense to that defined in [4] and uses the terminology of that document. The FEC Scheme provides FEC encoding and decoding and describes the protocol fields and or procedures used to identify packet payload data in the context of the FEC Scheme. The interface between the FEC Framework and an FEC Scheme, which is described in this document, is a logical one, which exists for specification purposes only. At an encoder, the FEC Framework passes groups of transport packet payloads to the FEC Scheme for FEC Encoding. The FEC Scheme returns FEC repair packet payloads, encoded FEC Payload ID information for each of the repair packets and, in some cases, encoded FEC Payload ID information for each of the source packets. At a decoder, the FEC Framework passes transport packet payloads (source and repair) to the FEC Scheme and the FEC Scheme returns additional recovered source packet payloads.

This document defines certain FEC Framework Configuration Information which MUST be available to both sender and receiver(s). For example, this information includes the specification of the transport flows which are to be FEC protected, specification of the transport flow(s) which will carry the FEC protection (repair) data and the relationship(s) between these 'source' and 'repair' flows (i.e. which source flow(s) are protected by each repair flow. The FEC Framework Configuration Information also includes information fields which are specific to the FEC Scheme. This information is analagous to the FEC Object Transmission Information defined in [4].

The FEC Framework does not define how the FEC Framework Configuration Information for the stream is communicated from sender to receiver. This must be defined by any Content Delivery Protocol specification as described below. However, this specification does define new Session Description Protocol (SDP) [5] elements which MAY be used by Content Delivery Protocols for this purpose.

The architecture outlined above is illustrated in the Figure 1.

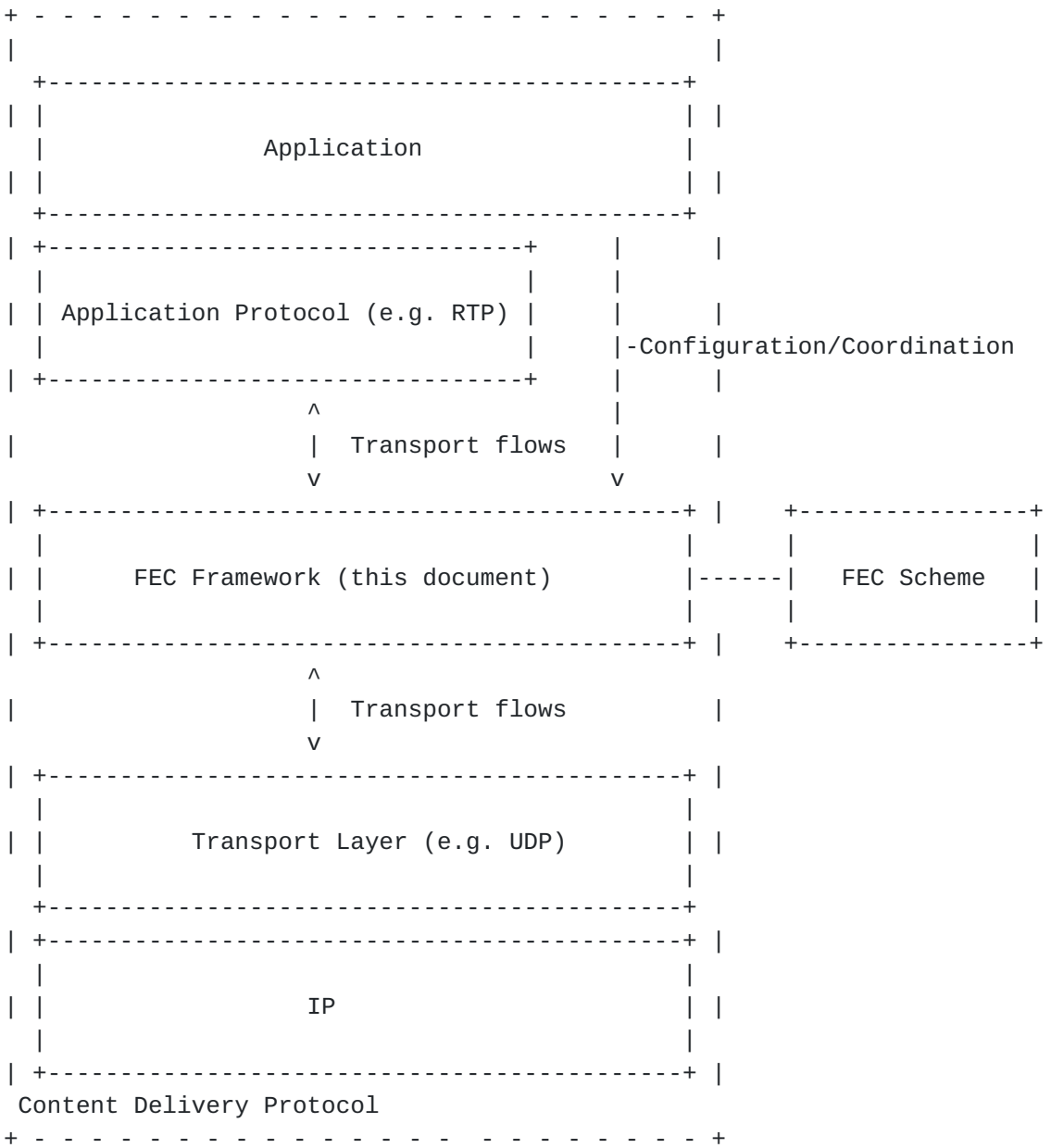

Figure 1: FEC Framework Architecture

5. Procedural overview

5.1. General

The mechanism defined in this document does not place any restrictions on the source data which can be protected together, except that the source data is carried over a supported transport protocol. The data may be from several different transport flows that are protected jointly. The FEC framework handles the packet flows as a sequence of 'source blocks' each consisting of a set of source packets, possibly from multiple flows which are to be protected together. For example, each source block may be constructed from those source packets related to a particular segment in time of the flow.

At the sender, the FEC Framework passes the packet payloads for all packets of a given block to the FEC Scheme for FEC encoding. The FEC Scheme performs the FEC encoding operation and returns the following information:

- optionally, encoded FEC Payload IDs for each of the source packets
- one or more FEC repair packet payloads
- encoded FEC Payload IDs for each of the repair packets

The FEC Framework then appends the FEC Payload IDs, if provided, to each of the source packets and sends the resulting packets, known as FEC SOurce Packets, to the receiver. The FEC repair packets are then constructed from the provided repair data and FEC Payload IDs and sent to the receiver. FEC repair packets are sent to a different transport port than the source packets, as specified by the FEC Configuration Information. In the case of multicast, FEC repair packets MAY be sent to a different multicast group or groups from the source packets.

This document does not define how the sender determines which source packets are included in which source blocks. A specific Content Delivery Protocol MAY define this mapping or it MAY be left as implementation dependent at the sender. However, a CDP specification MUST define how a receiver determines the length of time it should wait to receive FEC repair packets for any given source block.

The receiver recovers original source packets directly from any FEC Source packets received simply by removing the FEC Payload ID, if present. The receiver also passes the contents of the received FEC Source Packets, including their FEC Payload IDs to the FEC Scheme for decoding.

If any FEC Source packets related to a given source block have been lost, then the FEC Scheme may perform FEC decoding to recover the missing source packets (assuming sufficient FEC Source and FEC Repair packets related to that source block have been received).

Note that the receiver may need to buffer received source packets to allow time for the FEC Repair packets to arrive and FEC decoding to be performed before some or all of the received or recovered packets are passed to the application. If such a buffer is not provided, then the application must be able to deal with the severe re-ordering of packets that will be required. However, such buffering is Content Delivery Protocol and/or implementation-specific and is not specified here.

The FEC Source packets MUST contain information which identifies the source block and the position within the source block occupied by the packet. The identity of the source block and the position within the source block of a source packet are together known as the 'Source FEC Payload ID'. The FEC Scheme is responsible for defining and interpreting this information. This information MAY be encoded into a specific field within the FEC Source packet format defined in this specification, called the encoded Source FEC Payload ID field. The exact contents and format of the encoded Source FEC Payload ID field are defined by the FEC Scheme. Alternatively, the FEC Scheme or CDP MAY define how the Source FEC Payload ID is derived from other fields within the source packets. This document defines the way that the Source FEC Payload ID field is appended to source packets to form FEC Source packets.

The FEC Repair packets MUST contain information which identifies the source block and the relationship between the contained repair data and the original source block. This is known as the 'Repair FEC Payload ID'. This information MUST be encoded into a specific field, the Repair FEC Payload ID field, the contents and format of which are defined by the FEC Scheme.

Any FEC Schemes to be used in conjunction with this specification MUST be a systematic FEC Scheme. The FEC Scheme MAY use different encoded FEC Payload ID field formats for FEC Source packets and FEC Repair packets.

5.2. Sender Operation

It is assumed that the sender has constructed or received original data packets for the session. These may be RTP, RTCP, MIKEY or other UDP packets. The following operations describe a possible way to generate compliant FEC Source packet and FEC repair packet streams:

1. A source block is constructed as specified in Section 6.2.

2. The source block is passed to the FEC Scheme for FEC encoding. The Source FEC Payload ID information of each Source packet is determined by the FEC Scheme and, if necessary, encoded into encoded Source FEC Payload ID field.

3. The FEC Source packet is constructed according to Section 6.3. The identity of the original flow is maintained by the source packet through the use of the same transport ports and IP addresses which have been advertised by the Content Delivery Protocol (for example using SDP), as carrying FEC Source packets generated from an original stream of a particular protocol (e.g. RTP, RTCP, SRTP, MIKEY etc.). The FEC Source packet generated is sent according to normal transport layer procedures.

4. The FEC Scheme generates repair packet payloads from a source block and an encoded FEC Payload ID field for each repair paylaod. The FEC Framework places these payloads and FEC Payload IDs into FEC Repair packets, to be conveyed to the receiver(s). These repair packets are sent using normal transport layer procedures to a unique destination port(s) and/or multicast group(s) in the case of multicast to separate them from any of the source packet flows. The port(s) and multicast group(s) to be used for FEC Repair packets are defined in the FEC Framework Configuration Information.

5.3. Receiver Operation

The following describes a possible receiver algorithm, when receiving an FEC source or repair packet:

1. If an FEC Source packet is received (as indicated by the transport flow on which was received), the source packet and Source FEC Payload ID field are passed to the FEC Scheme.

2. If an FEC repair packet is received (as indicated by the transport flow on which it was received), the contained repair data and Repair FEC Payload ID field are passed to the FEC Scheme.

3. The FEC Scheme uses the received FEC Payload IDs to group source packets into source blocks.

4. If at least one source packet is missing from a source block, and at least one repair packet has been received for a source block then FEC decoding may be desirable. The FEC Scheme determines if enough data for decoding of any or all of the missing source packets in the source block has been received and,

if so, performs a decoding operation.

4. The FEC Scheme returns the source data to the FEC Framework in the form of source blocks containing received and decoded source packets and indications of any source packets which were missing and could not be decoded.

Note that the above procedure may result in a situation in which not all original source packets are recovered.

Source packets which are correctly received and those which are reconstructed MAY be delivered to the application out of order and in a different order from the order of arrival at the receiver. Alternatively, buffering and packet re-ordering MAY be required to re-order received and reconstructed source packets into the order they were placed into the source block, if that is necessary according to the application.

6. Protocol Specification

6.1. General

This section specifies the protocol elements for the FEC Framework. The protocol consists of three components which are described in the following sections:

1. Construction of a source block from source packets. The FEC code will be applied to this source block to produce the repair data.

2. A format for packets containing source data.

3. A format for packets containing repair data.

The operation of the FEC Framework is governed by certain FEC Framework Configuation Information. This configuration information is also defined in this section. A complete protocol specification that uses this framework MUST specify the means to determine and communicate this information between sender and receiver. Suitable Session Description Protocol elements for this purpose are defined in Section 8.

6.2. Structure of the source block

The FEC Framework and FEC Scheme exchange source data in the form of source blocks. A source block is generated from an ordered sequence of source packets. For each source packet, the following information is included in the source block:

- The identity of the transport flow on which the packet was recieved

- The original source packet payload

- The length of the original source packet payload

6.3. Packet format for FEC Source packets

The packet format for FEC Source packets MUST be used to transport the payload of an original source packet. As depicted in Figure 2, it consists of the original packet, optionally followed by the Source FEC Payload ID field. The FEC Scheme determines whether the Source FEC Payload ID field is required. This determination is specific to each transport flow.

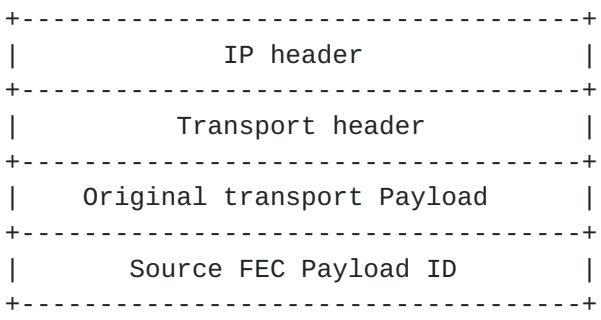

Figure 2: Structure of the FEC packet format for FEC Source packets

The IP and transport header fields MUST be identical to those of the original source packet. The Original transport Payload field MUST be identical to the transport payload of the original source packet. The transport payload of the FEC Source packet MUST consist of the Original Transport Payload followed by the Source FEC Payload ID field, if required.

The Source FEC Payload ID field contains information required to associate the source packet with a source block and for the operation of the FEC algorithm and defined by the FEC Scheme. The format of the Source FEC Payload ID field is defined by the FEC Scheme. Note that in the case that the FEC Scheme or CDP defines a means to derive the Source FEC Payload ID from other information in the packet (for example the a sequence number of some kind used by the application protocol), then the Source FEC Payload ID field is not included in the packet. In this case the original source packet and FEC Source Packet are identical.

Note: The Source FEC Payload ID is placed at the end of the packet so that in the case that Robust Header Compression [3] or other header compression mechanisms are used and in the case that a ROHC profile is defined for the protocol carried within the transport payload (for example RTP), then ROHC will still be applied for the FEC Source packets.

6.4. Packet Format for FEC Repair packets

The packet format for FEC Repair packets is shown in Figure 3. The transport payload consists of a Repair FEC Payload ID field followed by repair data generated in the FEC encoding process.

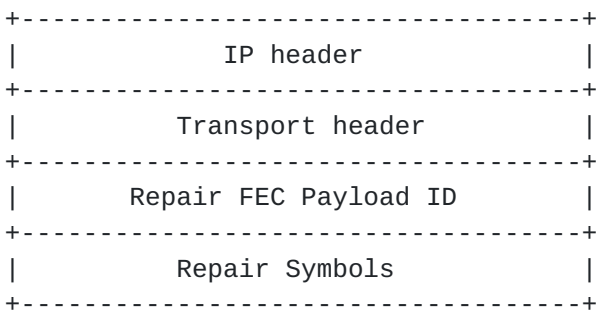

Figure 3: Packet format for repair packets

The Repair FEC Payload ID field contains information required for the operation of the FEC algorithm. This information is defined by the FEC Scheme. The format of the Repair FEC Payload ID field is defined by the FEC Scheme.

6.5. FEC Framework Configuration Information

The FEC Framework Configuration Information is information that the FEC Framework needs in order to apply FEC protection to the trasport flows. A complete Content Delivery Protocol specification that uses the framework specified here MUST include details of how this information is derived and communicated between sender and receiver.

The FEC Framework Configuration Information includes identification of a number of packet flows. For example, in the case of UDP, each packet flow is uniquely identified by a tuple { Source IP Address, Destination IP Address, Source UDP port, Destination UDP port }.

A single instance of the FEC Framework provides FEC protection for all packets of a specified set of source packet flows, by means of one or more packet flows consisting of repair packets. The FEC Framework Configuation Information includes, for each instance of the FEC Framework:

1. Identification of the packet flow(s) carrying FEC Repair packets, known as the FEC repair flow(s).

2. For each source packet flow protected by the FEC repair flow(s):

   a. Identification of the packet flow carrying source packets.

   b. An integer identifier, between 0 and 255, for this flow. This identifier MUST be unique amongst all source packet flows which are protected by the same FEC repair flow.

3. The FEC Encoding ID, identifying the FEC Scheme

4. An opaque container for FEC-Scheme-specific information

Multiple instances of the FEC Framework, with separate and independent FEC Framework Configuration Information, may be present at a sender or receiver. A single instance of the FEC Framework protects all packets of all the source packet flows identified in (2) above i.e. all packets on those flows MUST be FEC Source packets as defined in Section 6.3. A single source packet flow MUST NOT be protected by more than one FEC Framework instance.

A single FEC repair flow provides repair packets for a single instance of the FEC Framework. Other packets MUST NOT be sent within this flow i.e. all packets in the FEC repair flow MUST be FEC repair packets as defined in Section 6.4 and MUST relate to the same FEC Framework instance.

6.6. FEC Scheme requirements

In order to be used with this framework, an FEC Scheme MUST:

- use a systematic FEC code

- be based on discrete source blocks

Editor's note: This section requires expansion to define more explicitly the things an FEC Scheme must specify, along the lines of the FEC Building Block.

7. Transport Protocols

The following transport protocols are supported:

- User Datagram Protocol (UDP)
- Datagram Congestion Control Protocol (DCCP)

Editor's note: This section will contain transport-specific considerations, if any.

8. Session Description Protocol elements

This section defines Session Descrption Protocol elements which MAY be used by Content Delivery Protocols that make use of this framework to communicate the FEC Framework Configuration Information.

> NOTE: It is for further discussion whether these SDP elements should be defined here or in the context of a specific and complete Content Delivery Protocol specification for streaming.

This specification defines a class of new Transport Protocol identifiers for use in SDP media descriptions. For all existing identifiers <proto> this specification defines the identifier 'udp/fec/<proto>'. This identifier may be used as the Transport Protocol identifier for a media description for source data to indicate that the FEC Source packet format defined in Section 6.3 is used, with the original transport payload field formated according to <proto>.

Note that in the case of an FEC Scheme in which the Source FEC Payload ID field is not used, then the original Transport Protocol identifier MAY be used to support interoperability with receivers which do not support FEC at all, whilst also providing FEC protection for those receivers which support it.

A further Transport Protocol identifier, 'udp/fec', is defined to indicate the the FEC Repair Packet format defined in Section 6.4.

This specification describes the use of SDP attributes defined in [6] and the FEC grouping semantics defined in [7] to provide the FEC Framework Configuration Information. The 'fec-declaration' attribute may be used at either the session or media layer to declare a local identifier for a set of FEC parameters. This local identifier can then be referenced in the other attributes. This avoids duplication of parameter declarations within the SDP. The 'fec' parameter is used on the media level to associate a media description with a previous FEC parameter declaration. Finally, the 'FEC' grouping attribute semantics is used to associate together source and repair flows and assign UDP flow identifiers to be used in the source block construction.

Mechanisms for communicating the corresponance between source flows and the Flow Identifiers require further discussion.

8.1. udp/fec/<proto> transport protocol identifier

tbc

8.2. udp/fec transport protocol identifier

tbc

8.3. fec-declaration attribute

See [6].

8.4. fec-oti-extension attribute

See [6].

8.5. fec attribute

See [6].

8.6. FEC media grouping semantics

This attribute is used to group source flows and the single repair flow that protects them as described in [7] with the following additional requirements:

The media components grouped by an instance of the FEC grouping attribute MUST include exactly one component with the udp/fec protocol identifier.

The media components grouped by an instance of the FEC grouping attribute MUST include at least one and MAY include more than one source media stream with protocol identifier udp/fec/<proto>, where <proto> is a valid protocol identifier registered with IANA.

In the case of an FEC Scheme which defines an FEC Payload ID field of zero length, then the media components grouped by an instance of the FEC grouping attribite MAY include source media streams with protocol identified udp/<proto>, where <proto> is a valid protocol identifier registered with IANA.

8.7. SDP example

tbc

9. Congestion Control

This section starts with a informative section on the motivation of the normative requirements for congestion control, which are spelled out in Section 9.1.

Informative Note: The enforcement of Congestion Control (CC) principles has gained a lot of momentum in the IETF over the recent years. While the need of CC over the open Internet is unquestioned, and the goal of TCP friendliness is generally agreed for most (but not all) applications, the subject of congestion detection and measurement in heterogenous networks can hardly be considered as solved. Most congestion control algorithms detect and measure congestion by taking (primarily or exclusively) the packet loss rate into account. This appears to be inappropriate in environments where a large percentage of the packet losses are the result link-layer errors and independent of the network load. Note that such environments exist in the "open Internet", as well as in "closed" IP based networks. An example for the former would be the use of IP/UDP/RTP based streaming from an Internet-connected streaming server to a device attached to the Internet using cellular technology.

The authors of this draft are primarily interested in applications where the application reliability requirements and end-to-end reliability of the network differ, such that it warrants higher layer protection of the packet stream - for example due to the presence of unreliable links in the end-to-end path - and where real-time, scalability or other constraints prohibit the use of higher layer (transport or application) feedback. A typical example for such applications is multicast and broadcast streaming or multimedia transmission over heterogenous networks. In other cases, application reliability requirements may be so high that the required end-to-end reliability is difficult to achieve even over wired networks. Furthermore the end-to-end network reliability may not be known in advance.

This FEC framework is not proposed, nor intended, as a QoS enhancement tool to combat losses resulting from highly congested networks. It should not be used for such purposes.

In order to prevent such mis-use, standardization could be left to bodies most concerned with the problem described above. However, the IETF defines base standards used by several bodies, including DVB, 3GPP, 3GPP2, all of which appear to share the environment and the problem described.

Alternatively, a clear applicability statement could be used - for example restricting use of the framework to networks with wireless links. However, there may be applications where the use of FEC may be justified to combat congestion-induced packet losses - particularly in lightly loaded networks, where congestion is the result of relatively rare random peaks in instantaneous traffic load - thereby intentionally violating congestion control principles. One possible example for such an application could be a no-matter-what, brute-force FEC protection of traffic generated as an emergency signal.

We propose a third approach, which is to require at a minimum that the use of this framework with any given application, in any given environment, does not cause congestion issues which the application alone would not itself cause i.e. the use of this framework must not make things worse.

Taking above considerations into account, the normative text of this section implements a small set of constraints for the FEC, which are mandatory for all senders compliant with this FEC framework. Further restrictions may be imposed for certain Content Delivery Protocols. In this it follows the spirit of the congestion control section of RTP and its Audio-Visual Profile (RFC3550/STD64 and RFC3551/STD65).

One of the constraints effectively limits the bandwidth for the FEC protected packet stream to be no more than roughly twice as high as the original, non-FEC protected packet stream. This disallows the (static or dynamic) use of excessively strong FEC to combat high packet loss rates, which may otherwise be chosen by naively implemented dynamic FEC-strength selection mechanisms. We acknowledge that there may be a few exotic applications, e.g. IP traffic from space-based senders, or senders in certain hardened military devices, which would warrant a higher FEC strength. However, in this specification we give preference to the overall stability and network friendliness of the average application, and for those a factor of 2 appears to be appropriate.

A second constraint requires that the FEC protected packet stream be in compliance with the congestion control in use for the application and network in question.

9.1. Normative requirements

The bandwidth of FEC Repair packet flows MUST NOT exceed the bandwidth of the source packet flows being protected. In addition, whenever the source packet flow bandwidth is adapted due to the operation of congestion control mechanisms, the FEC repair packet

flow bandwidth MUST be similarly adapted.

10. Security Considerations

The application of FEC protection to a stream does not provide any kind of security protection.

If security services are required for the stream, then they MUST either be applied to the original source data before FEC protection is applied, or to both the source and repair data, after FEC protection has been applied.

If integrity protection is applied to source packets before FEC protection is applied, and no further integrity protection is applied to repair packets, then a denial of service attack is possible if an attacker is in a position to inject fake repair packets. If received by a receiver, such fake repair packets could cause incorrect FEC decoding resulting in incorrect source packets being passed up to the application protocol. Such incorrect packets would then be detected by the source integrity protection and discarded, resulting in partial or complete denial of service. Therefore, in such environments, integrity protection MUST also be applied to the FEC Repair packets, for example using IPsec. Receivers MUST also verify the integrity of source packets before including the source data into the source block for FEC purposes.

It is possible that multiple streams with different confidentiality requirements (for example, the streams may be visible to different sets of users) can be FEC protected by a single repair stream. This scenario is not recommended, since resources will be used to distribute and decode data which cannot then be decrypted by at least some receivers. However, in this scenario, confidentiality protection MUST be applied before FEC encoding of the streams, otherwise repair data may be used by a receiver to decode unencrypted versions of source streams which they do not have permissionions to view.

11. IANA Considerations

tbc

12. Acknowledgments

This document is based in large part on [8] and so thanks are due to the additional authors of that document, Mike Luby, Magnus Westerlund and Stephan Wenger. That document was in turn based on the FEC streaming protocol defined by 3GPP in [9] and thus thanks are also due to the participants in 3GPP TSG SA working group 4.

13. References

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

[2] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[3] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.

[4] Watson, M., "Forward Error Correction (FEC) Building Block", draft-ietf-rmt-fec-bb-revised-04 (work in progress), September 2006.

[5] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[6] Mehta, H., "SDP Descriptors for FLUTE", draft-mehta-rmt-flute-sdp-05 (work in progress), January 2006.

[7] Li, A., "Forward Error Correction Grouping Semantics in Session Description Protocol", RFC 4756, November 2006.

[8] Watson, M., "Forward Error Correction (FEC) Streaming Framework", draft-watson-tsvwg-fec-sf-00 (work in progress), July 2005.

[9] 3GPP, "Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs", 3GPP TS 26.346, April 2005.

Author's Address

Mark Watson
Digital Fountain
39141 Civic Center Drive
Suite 300
Fremont, CA 94538
U.S.A.

Email: mark@digitalfountain.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).