Transport Area Working Group
Internet-Draft
Expires: January 8, 2006

M. Watson
M. Luby
Digital Fountain
M. Westerlund
Ericsson
S. Wenger
Nokia

July 7, 2005

Forward Error Correction (FEC) Streaming Framework
draft-watson-tsvwg-fec-sf-00

Status of this Memo

Copyright Notice

Abstract

This document defines a framework for applying Forward Error Correction to UDP flows, primarily intended for streaming media. This framework can be used to define Content Delivery Protocols, in the context of the RMT FEC Building Block, that provide Forward Error

Correction for streaming media delivery. Content Delivery Protocols defined using this framework can support FEC Schemes (and associated FEC codes) compliant with the FEC Building Block. Thus, Content Delivery Protocols can be defined which are not specific to a particular FEC Scheme.

Table of Contents

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) which make use of separately defined FEC Schemes. Any CDP defined according to the requirements of this building block can then easily be used with any FEC Scheme which is also defined according to the requirements of the FEC building block (Note that it is also possible that CDPs define additional requirements on the FEC Scheme. Such CDPs can clearly only be used with FEC Schemes compliant with those requirements.)

This document defines a framework for the definition of CDPs, in the sense of the FEC Building Block, which provide for FEC protection of streamed data flows over 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 Content Delivery Protocols that support streaming data over UDP.

The framework defined in this document is not specific to a single streaming application protocol. The framework provides FEC protection for application protocol flows over UDP and for combined protection of multiple such flows. For example, multiple RTP flows may be protected together with the associated RTCP flows and potentially also other related flows such as MIKEY packets. For many FEC Schemes in many loss conditions, the improvement in reliability achievable through the use of FEC with a given FEC overhead increases as the amount of data protected as a single block increases. Thus there is considerable advantage in the ability to protect multiple streams together, particularly in cases where the receiver requires all the streams in order to offer a useful service to the user.

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 Streaming Framework at

sender and receiver - for example details of the flows to be FEC protected and the flow(s) that will carry the FEC protection data. We also specify SDP [5] attributes which a Content Delivery Protocol MAY use to communicate this information.

2. Definitions/Abbreviations

Source Block: A logical block of data constructed from some subset of the source packets of the application protocol flows to which the FEC protection is to be applied.

FEC: Forward Error Correction. See [4].

Symbol: A unit of data processed by the Forward Error Correction code. A symbol is always considered as a unit i.e. it is either completely received or completely lost.

Source symbol: A symbol of a source block.

Repair symbol: A symbol containing information generated by the FEC code from a source block which can be used to recover lost source symbols from that source block.

Encoding symbol: A source symbol or a repair symbol.

Source Packet Information (SPI): Information related to or from a source packet which is included in the source block.

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

FEC Payload ID: See [4].

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.

FEC Object Transmission Information: See [4].

FEC Encoding ID: See [4].

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

The FEC-SF makes use of an FEC Scheme, in the sense of [4] and uses the terminology of that document. The FEC Scheme provides FEC encoding and decoding and describes the protocol fields used to identify packet payload data in the context of the FEC Scheme; i.e. it is the FEC Scheme specification, which MUST be defined according to [4], which defines the format and interpretation of the FEC Payload ID fields which are included in packets to identify the FEC source or repair symbol data which are carried by those packets. Alternatively, an FEC Scheme may define some other mechanism to identify the symbol data contained in a packet, in which case the FEC Payload ID fields described in the FEC Building Block and this specification may have zero length.

The FEC Streaming Framework does not define how the FEC Object Transmission Information for the stream is communicated from sender to receiver. This must be defined by any Content Delivery Protocol specification according to the requirements of the FEC Building Block. However, this specification does define new Session Description Protocol (SDP) [5] elements which MAY be used by Content Delivery Protocols for this purpose.

This document defines certain FEC Streaming Configuration Information which MUST be available to both sender and receiver(s). For example, this information includes the specification of the UDP flows which are to be FEC protected, specification of the UDP flow(s) which will carry the FEC protection (repair) data and the relationship(s) between these 'source' and 'repair' flows. The FEC Streaming Framework assumes that the Content Delivery Protocol includes appropriate signalling to communicate this FEC Streaming Configuration Information from sender to receiver(s). In many cases, Content Delivery Protocols may use SDP to communicate information about the UDP streams. This document defines suitable extensions to SDP which MAY be used to communicate the FEC Streaming Configuration Information from sender to receiver(s).

The architecture outlined above is illustrated in the Figure 1.

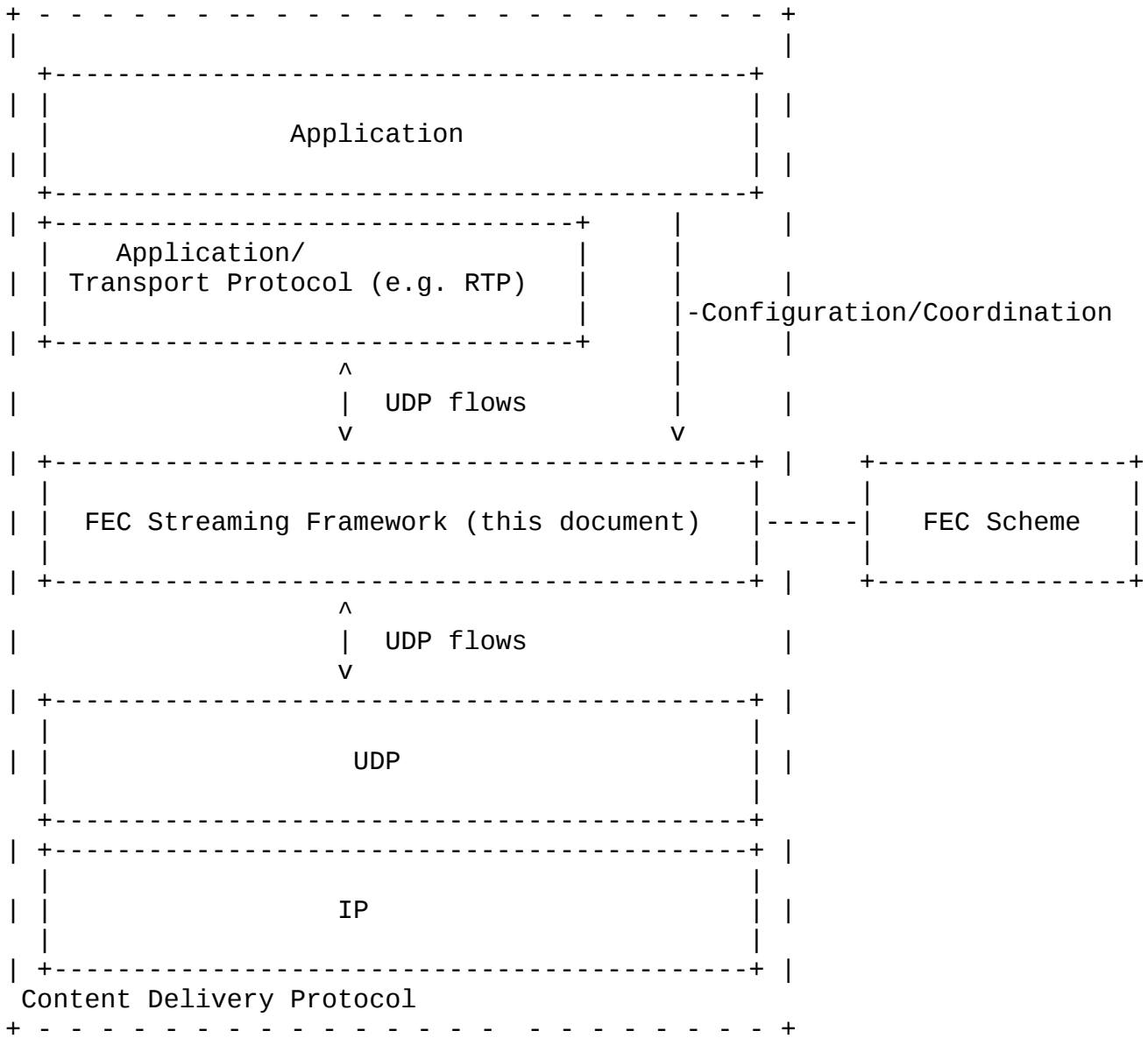

Figure 1: FEC Streaming Framework Architecture

5. Procedural overview

5.1 General

The mechanism defined in this document consists of three components:

(i) construction of a 'source block' from source media packets belonging to one or several UDP packet flows. The UDP flows MAY include, for example, RTP, RTCP and SRTP packets and also other protocols related to the stream such as MIKEY packets.

(ii) extension of source packets to indicate the source block and the position within the source block occupied by the data from an related to the source packet.

(iii) definition of repair packets, sent over UDP, which can be used by the FEC decoder to reconstruct missing portions of the source block.

The mechanism does not place any restrictions on the source data which can be protected together, except that the source data is carried over UDP. The data may be from several different UDP flows that are protected jointly. In general, multiple source blocks will be constructed for a stream each constructed from different sets of source packets. For example, each source block may be constructed from those source packets related to a particular segment of the stream in time.

A receiver supporting this streaming framework MUST support the packet format for FEC Source packets and MAY also support the packet format for FEC Repair 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.

At the sender, the mechanism processes original UDP packets to create:

(i) a stored copy of the original packets in the form of one or more 'source block(s)'. The source block is a logical block of data to which the the FEC code will subsequently be applied. It is constructed by concatenating 'Source Packet Information' (SPI) for each source packet. The SPI for a packet contains a short identifier for the flow the packet belongs to, the length of the

packet, the UDP payload and possible padding bytes.

(ii) FEC Source packets for transmission to the receiver.

The FEC Streaming Framework uses the FEC encoder specified by the FEC Scheme in use to generate the desired quantity of repair symbols from a source block. These repair symbols are then sent using the FEC repair packet format to the receiver. The FEC Repair packets are sent to a UDP destination port different from any of the original UDP packets' destination port(s) as indicated by the FEC Streaming Configuation Information.

The receiver recovers original source packets directly from any FEC Source packets received. The receiver also uses the received FEC Source Packets to construct a stored copy of the original packets in the same source block format as constructed at the sender.

If any FEC Source packets related to a given source block have been lost, then this copy of the source block at the receiver will be incomplete. If sufficient FEC source and FEC Repair packets related to that source block have been received, the FEC Framework may use the FEC decoding algorithm defined by the FEC Scheme to recover a (hopefully, but not necessarily, complete) copy of the source block. The SPI for the missing source packets can then be extracted from the completed parts of the source block and used to reconstruct the source packets to be passed to the application.

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 SPI derived from the packet. The identity of the source block and the position within the source block occupied by the SPI for a source packet are together known as the 'Source FEC Payload ID'. This information MAY be encoded into a specific field within the FEC Source packet format defined in this specification, called the Source FEC Payload ID field. The exact contents and format of the 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 and MUST be based on source blocks. The FEC Scheme MAY use different 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, by concatenating the SPI for each original source packet. In doing so, the Source FEC Payload ID information of the FEC Source packet can be determined and included in the Source FEC Payload ID field, if defined. In the SPI the identity of the packet's UDP flow is marked using a short 'UDP flow ID', defined in this specification. The association of UDP flow specifications to UDP flow IDs is defined by the FEC Streaming Configuation Information.

2. 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 UDP 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 UDP procedures.

3. The FEC encoder generates repair symbols from a source block and the FEC Streaming Framework places these symbols into FEC Repair packets, to be conveyed to the receiver(s). These repair packets are sent using normal UDP procedures to a unique destination port to separate them from any of the source packet flows. The ports to be used for FEC Repair packets are defined in the FEC Streaming 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 UDP flow on which was received):

   a. The original source packet is reconstructed by removing the Source FEC Payload ID. The resulting packet MAY be buffered to allow time for the FEC repair.

   b. The SPI for the resulting packet is placed into the source block according to the Source FEC Payload ID and the source block format described in Section 6.2. The IP addresses and UDP ports the packet was received on/sent from are used to determine the UDP flow ID within the SPI.

2. If an FEC repair packet is received (as indicated by the UDP flow on which it was received), the contained repair symbols are associated with a source block according to the Repair FEC Payload ID.

3. If at least one source packet is missing and at least one repair packet has been received for a source block then FEC decoding may be desirable. The FEC decoder determines if the source block constructed in step 1 plus the associated repair symbols received in step 2 contain enough symbols for decoding of any or all of the missing source symbols in the source block and, if so, performs a decoding operation.

4. Any SPI that was reconstructed during the decoding operation is then used to reconstruct the missing source packets and these are buffered as normal received source packets (see step 1a above).

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 Streaming 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 Streaming Framework is governed by certain FEC Streaming 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 7.

6.2 Structure of the source block

This clause defines the layout of the source block. The source block consists of concatenation of SPI for at least one original source UDP packet.

Let

n be the number of UDP packets in the source block. n MAY be determined dynamically during the source block construction process.

T be the source symbol size in bytes. Note: this information is provided by the FEC Scheme as defined in Section 6.6.

R[i] denote the octets of the UDP payload of the i-th UDP packet to be added to the source block, 0 <= i < n.

l[i] be the length of R[i] in octets

L[i] denote two octets representing the value of l[i] in network byte order (high order octet first)

f[i] denote an integer 'UDP flow ID' identifying the UDP flow from which the i-th packet was taken

F[i] denote a single octet representing the value of f[i]

s[i] be the smallest integer such that s[i]*T >= (l[i]+3). Note s[i] is the length of SPI[i] in units of symbols.

P[i] denote s[i]*T-(l[i]+3) zero octets. Note: P[i] are padding octets to align the start of each UDP packet with the start of a symbol.

SPI[i] be the concatenation of F[i] ,L[i], R[i] and P[i].

Then, the source block is constructed by concatenating SPI[i] for i = 0, 2, ... n-1. The source block size, S, is then given by sum {s[i]*T, i=0, ..., n-1}.

Source blocks are identified by integer Source Block Numbers and symbols within a source block by integer Encoding Symbol IDs. This specification does not specify how Source Block Numbers are allocated to source blocks. Symbols are numbered consecutively starting from zero within the source block. Each source packet is associated with the Encoding Symbol ID of the first symbol containing SPI for that packet. Thus, the Encoding Symbol ID value associated with the j-th source packet, ESI[j], is given by

ESI[j] = 0, for j=0

ESI[j] = sum{s[i], i=0,...,(j-1)}, for 0 < j < n

A UDP flow is uniquely defined by an IP source and destination address and UDP source and destination port values. The assignment of UDP flow ID values to UDP flows is part of the FEC Streaming Configuration Information.

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 UDP packet. As depicted in Figure 2, it consists of the original UDP packet, followed by the Source FEC Payload ID field.

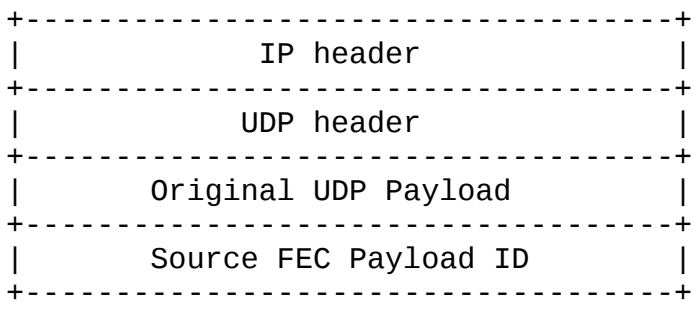

Figure 2: Structure of the FEC packet format for FEC Source packets

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

The Source FEC Payload ID field contains information required 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 RTP Sequence number), then the Source FEC Payload ID field described here may have zero length.

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 UDP 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 UDP payload consists of a Repair FEC Payload ID field and one or more repair symbols generated by the FEC encoding process.

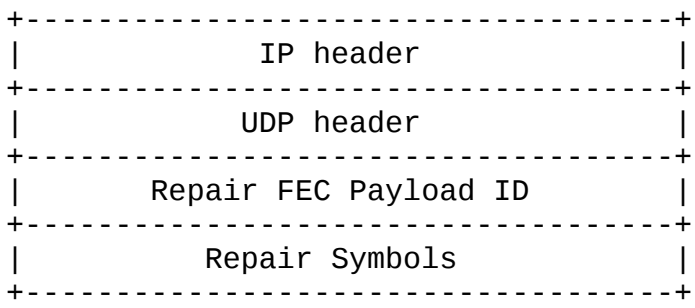

Figure 3: Packet format for repair packets

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.

Any number of whole repair symbols may be contained within an FEC Repair packet, subject to packet size restrictions or other restrictions defined by the FEC Scheme. The number of repair symbols within a packet can be determined from the symbol length and the packet length. Partial repair symbols MUST NOT be included in FEC repair packets.

6.5 FEC Streaming Configuration Information

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

The FEC Streaming Configuration Information includes identification of a number of UDP packet flows. Each UDP 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-SF provides FEC protection for all packets of a specified set of source UDP packet flows, by means of one or more UDP packet flows containing repair packets. The FEC Streaming Configuation Information includes, for each instance of the FEC-SF:

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

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

   a. Identification of the UDP packet flow carrying source packets.

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

3. The FEC Encoding ID, FEC Instance ID (if applicable) and, optionally, the symbol size.

Item (3) above is included in the FEC Object Transmission Information.

Multiple instances of the FEC-SF, with separate and independent FEC Streaming Configuration Information, may be present at a sender or receiver. A single instance of the FEC-SF protects all packets of all the source UDP 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 UDP packet flow MUST NOT be protected by more than one FEC-SF instance.

A single FEC repair flow provides repair packets for a single instance of the FEC-SF. 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-SF instance.

The FEC-SF requires to be informed of the symbol size to be used for each source block. This information MAY be included in the FEC Streaming Configuration Information or it MAY be communicated by other means, for example within the FEC Repair Payload ID field. A complete Content Delivery Protocol specification MUST specify how this information is communicated between sender and receiver.

6.6 FEC Scheme requirements

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

- adhere to the requirements of [4]

- be systematic

- be based on discrete source blocks

- specify how the Source Block Number and Encoding Symbol ID associated with a source packet are derived or communicated from sender to receiver (for example, within the Source FEC Payload ID field)

- specify how the symbol length is derived or communicated from sender to receiver (for example, as part of the FEC Object Transmission Information).

7. 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 Streaming 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 UDP payload field formated according to <proto>.

Note that in the case of an FEC Scheme in which the Source FEC Payload ID has zero length, then the original Transport Protocol identifier MAY be used to support interoperability with devices which do not support the FEC Source packet format at all, whilst also providing FEC protection for those devices 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 Streaming 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 UDP Flow Identifiers that are included within the source block require further discussion.

7.1 udp/fec/<proto> transport protocol identifier

tbc

7.2 udp/fec transport protocol identifier

tbc

7.3 fec-declaration attribute

See [6].

7.4 fec-oti-extension attribute

See [6].

7.5 fec attribute

See [6].

7.6 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.

7.7 SDP example

tbc

8. 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 8.1.

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 or other constraints prohibit the use of higher layer (transport or application) feedback. A typical example for such applications is multicast and broadcast streaming to wireless devices or multimedia transmission over heterogenous, but partly wireless 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-H, 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.

8.1 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.

9. 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.

10. IANA Considerations

tbc

11. Acknowledgments

This framework is based in large part on the FEC streaming protocol defined by 3GPP in [8] and thus thanks are due to the participants in 3GPP TSG SA working group 4.

12. 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-00 (work in progress), May 2005.

[5] Handley, M., "SDP: Session Description Protocol", draft-ietf-mmusic-sdp-new-24 (work in progress), February 2005.

[6] Mehta, H., "SDP Descriptors for FLUTE", draft-mehta-rmt-flute-sdp-03 (work in progress), July 2005.

[7] Li, A., "FEC Grouping Semantics in SDP", draft-li-mmusic-fec-grouping-00 (work in progress), June 2005.

[8] 3GPP, "Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs", 3GPP TS 26.346 6.1.0, April 2005.

Authors' Addresses

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

Email: mark@digitalfountain.com

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

Email: luby@digitalfountain.com

Magnus Westerlund
Ericsson
Ericsson Research
SE-164 80 Stockholm
SWEDEN

Email: magnus.westerlund@ericsson.com

Stephan Wenger
Nokia

Email: Stephan.Wenger@nokia.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.