

RMCAT WG  
Internet-Draft  
Intended status: Experimental  
Expires: January 5, 2015

V. Singh  
M. Nagy  
J. Ott  
Aalto University  
L. Eggert  
NetApp  
July 4, 2014

Congestion Control Using FEC for Conversational Media  
draft-singh-rmcat-adaptive-fec-00

Abstract

This document describes a new mechanism for conversational multimedia flows. The proposed mechanism uses Forward Error Correction (FEC) encoded RTP packets (redundant packets) along side media packets to probe for available network capacity. A straightforward interpretation is, the sending endpoint increases the sending rate by keeping the media rate constant but increases the amount of FEC. If no losses and discards occur, the endpoint can then increase the media rate. If losses occur, the redundant FEC packets help in recovering the lost packets. Consequently, the endpoint can vary the FEC bit rate to conservatively (by a small amount) or aggressively (by a large amount) probe for available network capacity.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on January 5, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	3
3. Concept: FEC for Congestion Control . . . . .	4
3.1. State Machine . . . . .	5
3.2. FEC Scheme . . . . .	5
3.3. Applicability to other RMCAT Schemes . . . . .	5
4. Security Considerations . . . . .	5
5. IANA Considerations . . . . .	5
6. Acknowledgements . . . . .	5
7. References . . . . .	6
7.1. Normative References . . . . .	6
7.2. Informative References . . . . .	7
Appendix A. Simulations . . . . .	7
Authors' Addresses . . . . .	8

## 1. Introduction

The Real-time Transport Protocol (RTP) [RFC3550] is widely used in voice telephony and video conferencing systems. Many of these systems run over best-effort UDP/IP networks, and are required to implement congestion to adapt the transmission rate of the multimedia streams to match the available network capacity, while maintaining the user-experience [I-D.ietf-rmcat-cc-requirements]. The circuit breakers [I-D.ietf-avtcore-rtp-circuit-breakers] describe a minimal set of conditions when an RTP media stream is causing severe congestion and should cease transmission. Consequently, the congestion control algorithm are expected to avoid triggering these conditions.

Conversational multimedia systems use Negative Acknowledgment (NACK), Forward Error Correction (FEC), and Reference Picture Selection (RPS) to protect against packet loss. These are used in addition to the codec-dependent resilience methods (for e.g., full intra-refresh and error-concealment). In this way, the multimedia system is anyway trading off part of the sending rate for redundancy or retransmissions to reduce the effects of packet loss. An endpoint

often prefers using FEC in high latency networks where retransmissions may arrive later than the playout time of the packet (due to the size of the dejitter buffer) [Holmer13]. Therefore, the endpoint needs to adapt the transmission rate to best fit the changing network capacity and the amount of redundancy based on the observed/expected loss rate and network latency. Figure 1 shows the applicability of different error-resilience schemes based on the end-to-end latency and the observed packet loss [Devadoss08].

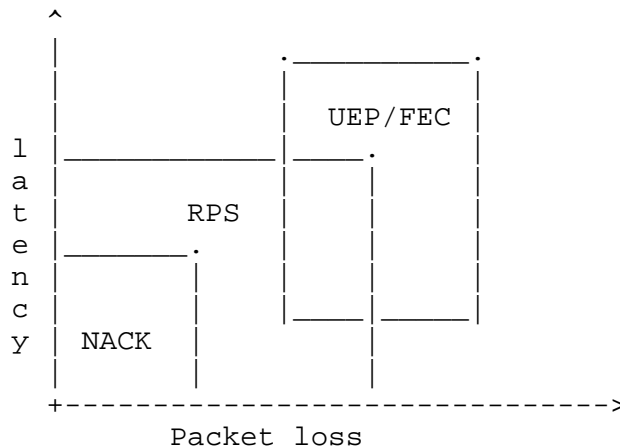


Figure 1: Applicability of Error Resilience Schemes based on the network delay and observed packet loss

In this document, we describe the use of FEC packets not only for error-resilience but also as a probing mechanism for congestion control (ramping up the transmission rate).

## 2. Terminology

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 BCP 14, [RFC2119] and indicate requirement levels for compliant implementations.

The terminology defined in RTP [RFC3550], RTP Profile for Audio and Video Conferences with Minimal Control [RFC3551], RTCP Extended Report (XR) [RFC3611], Extended RTP Profile for RTCP-based Feedback (RTP/AVPF) [RFC4585], RTP Retransmission Payload Format [RFC4588], Forward Error Correction (FEC) Framework [RFC6363], and Support for Reduced-Size RTCP [RFC5506] apply.

### 3. Concept: FEC for Congestion Control

FEC is one method for providing error-resilience, it improves reliability by adding redundant data to the primary media flow, which is used by receiver to recover packets that have been lost due to congestion or bit-errors. The congestion control algorithm on the other hand aims at maximizing the network path utilization, but risks over-estimating the available end-to-end network capacity leading to congestion (and therefore losses).

The main idea behind using FEC for congestion control is as follows: the sending endpoint chooses a high FEC rate to aggressively probe for available capacity and conversely chooses a low FEC rate to conservatively probe for available capacity. During the ramp up, if a packet is lost and the FEC packet arrives in time for decoding, the receiver is able to recover the lost packet; if no packet is lost, the sender is able to increase the media encoding rate by swapping out a part of the FEC rate. This method can be especially useful when the sending rate is close to the bottleneck link rate: by choosing an appropriate FEC rate, the endpoint is able to probe for available capacity without changing the target media rate and therefore not affecting the user-experience. Therefore, the congestion control algorithm is always able to probe for available capacity, as improved reliability compensates for possible errors resulting from overuse (i.e., increase in observed latency and/or losses).

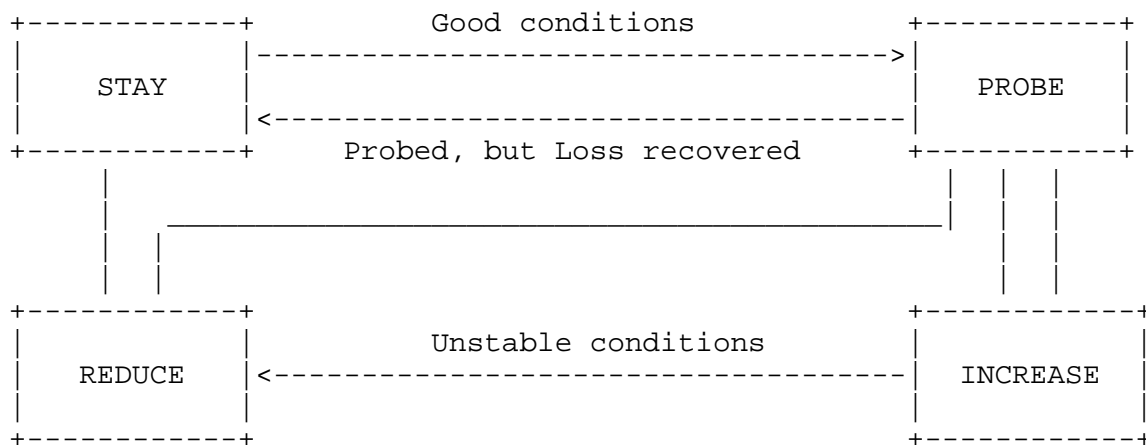


Figure 2: State machine of a Congestion Control enabling FEC.

### 3.1. State Machine

Figure 2 illustrates the the state machine of a congestion control algorithm incorporating FEC for probing. The state-machine includes 4 states: STAY, PROBE, INCREASE, REDUCE.

### 3.2. FEC Scheme

[RFC6363] describes a framework for using Forward Error Correction (FEC) codes with RTP and allows any FEC code to be used with the framework. For this proposal, the FEC packets are created by XORing RTP media packets, the resulting redundant RTP packets are encoded using the scheme defined in [RFC5109].

The endpoint MAY use a single-frame FEC or a multi-frame FEC for protecting the media stream. A single-frame FEC protects against a single packet loss and fails when burst loss occurs. Using multi-frame FEC helps mitigate these issues at the cost of higher overhead and latency in recovering lost packets. [Holmer13] shows examples of using a single- and multi-frame FEC.

The receiving endpoint may report the post-repair loss (or residual loss) using either the report block defined in [RFC5725] (Run-length encoding of packets repaired) or in [I-D.ietf-xrblock-rtcp-xr-post-repair-loss-count] (packet count of repaired packets).

### 3.3. Applicability to other RMCAT Schemes

[Open issue: The current implementation is delay based and is documented in [Nagy14]. However, we would like to generalize the concept and apply it to different RMCAT algorithms for e.g., Google's Congestion Control algorithm [I-D.alvestrand-rmcat-congestion], etc.]

## 4. Security Considerations

TBD

## 5. IANA Considerations

There are no IANA impacts in this memo.

## 6. Acknowledgements

This document is based on the results published in [Nagy14], it has benefited from several rounds of peer-review and hallway discussions.

## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.
- [RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.
- [RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.
- [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.
- [I-D.ietf-avtcore-rtp-circuit-breakers]  
Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", draft-ietf-avtcore-rtp-circuit-breakers-05 (work in progress), February 2014.
- [RFC5109] Li, A., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, December 2007.
- [RFC5725] Begen, A., Hsu, D., and M. Lague, "Post-Repair Loss RLE Report Block Type for RTP Control Protocol (RTCP) Extended Reports (XRs)", RFC 5725, February 2010.
- [I-D.ietf-xrblock-rtcp-xr-post-repair-loss-count]  
Huang, R. and V. Singh, "RTP Control Protocol (RTCP) Extended Report (XR) for Post-Repair Loss Count Metrics", draft-ietf-xrblock-rtcp-xr-post-repair-loss-count-05 (work in progress), June 2014.

## 7.2. Informative References

- [RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.
- [RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, October 2011.
- [I-D.ietf-rmcat-cc-requirements]  
Jesup, R., "Congestion Control Requirements For RMCAT", draft-ietf-rmcat-cc-requirements-02 (work in progress), February 2014.
- [I-D.alvestrand-rmcat-congestion]  
Holmer, S., Cicco, L., Mascolo, S., and H. Alvestrand, "A Google Congestion Control Algorithm for Real-Time Communication", draft-alvestrand-rmcat-congestion-02 (work in progress), February 2014.
- [I-D.sarker-rmcat-eval-test]  
Sarker, Z., Singh, V., Zhu, X., and M. Ramalho, "Test Cases for Evaluating RMCAT Proposals", draft-sarker-rmcat-eval-test-00 (work in progress), February 2014.
- [Nagy14] Nagy, M., Singh, V., Ott, J., and L. Eggert, "Congestion Control using FEC for Conversational Multimedia Communication", Proc. of 5th ACM International Conference on Multimedia Systems (MMSys 2014) , 3 2014.
- [Devadoss08]  
Devadoss, J., Singh, V., Ott, J., Liu, C., Wang, Y-K., and I. Curcio, "Evaluation of Error Resilience Mechanisms for 3G Conversational Video", Proc. of IEEE International Symposium on Multimedia (ISM 2008) , 3 2014.
- [Holmer13]  
Holmer, S., Shemer, M., and M. Paniconi, "Handling Packet Loss in WebRTC", Proc. of IEEE International Conference on Image Processing (ICIP 2013) , 9 2013.

## Appendix A. Simulations

This document is based on the results published in [Nagy14]. See the paper for ns-2 and testbed results; more results based on the scenarios listed in [I-D.sarker-rmcat-eval-test] will be published shortly.

## Authors' Addresses

Varun Singh  
Aalto University  
School of Electrical Engineering  
Otakaari 5 A  
Espoo, FIN 02150  
Finland

Email: [varun@comnet.tkk.fi](mailto:varun@comnet.tkk.fi)  
URI: <http://www.netlab.tkk.fi/~varun/>

Marcin Nagy  
Aalto University  
School of Electrical Engineering  
Otakaari 5 A  
Espoo, FIN 02150  
Finland

Email: [marcin.nagy@aalto.fi](mailto:marcin.nagy@aalto.fi)

Joerg Ott  
Aalto University  
School of Electrical Engineering  
Otakaari 5 A  
Espoo, FIN 02150  
Finland

Email: [jo@comnet.tkk.fi](mailto:jo@comnet.tkk.fi)

Lars Eggert  
NetApp  
Sonnenallee 1  
Kirchheim 85551  
Germany

Phone: +49 151 12055791  
Email: [lars@netapp.com](mailto:lars@netapp.com)  
URI: <http://eggert.org/>