Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2010

M. Westerlund
I. Johansson
Ericsson
C. Perkins
University of Glasgow
July 6, 2009

Explicit Congestion Notification (ECN) for RTP over UDP
draft-westerlund-avt-ecn-for-rtp-00

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on January 7, 2010.

Copyright Notice

Copyright (c) 2009 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 in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document specifies how explicit congestion notification (ECN)

can be used with RTP/UDP flows that use RTCP as feedback mechanism.

Table of Contents

1. Introduction 3
2. Conventions, Definitions and Acronyms 3
3. Discussion, Requirements, and Design Rationale 4
3.1. Requirements 5
3.2. Applicability 6
4. Use of ECN with RTP/UDP/IP 9
4.1. Negotiation of ECN Capability 11
4.1.1. Signalling ECN Capability using SDP 11
4.1.2. ICE Parameter to Signal ECN Capability 12
4.2. Initiation of ECN Use in an RTP Session 12
4.2.1. Detection of ECT using RTP and RTCP 13
4.2.2. Detection of ECT using STUN with ICE 15
4.3. Ongoing Use of ECN Within an RTP Session 17
4.3.1. Transmission of ECT-marked RTP Packets 17
4.3.2. Reporting ECN Feedback via RTCP 17
4.3.3. Response to Congestion Notifications 18
4.4. Detecting Failures and Receiver Misbehaviour 20
4.4.1. Fallback mechanisms 21
5. RTCP Extension for ECN feedback 22
6. Processing RTCP ECN Feedback in RTP Translators and Mixers . . 24
6.1. Fragmentation and Reassembly in Translators 25
6.2. Generating RTCP ECN Feedback in Translators 25
6.3. Generating RTCP ECN Feedback in Mixers 25
7. Implementation considerations 26
8. IANA Considerations 26
8.1. SDP Attribute Registration 26
8.2. AVPF Transport Feedback Message 26
8.3. STUN attribute 26
8.4. ICE Option 27
9. Security Considerations 27
10. References 29
10.1. Normative References 29
10.2. Informative References 30
Authors' Addresses 32

1. Introduction

This document outlines how Explicit Congestion Notification (ECN) [RFC3168] can be used for RTP [RFC3550] flows running over UDP/IP which use RTCP as feedback mechanism. The solution consists of feedback of ECN congestion experienced markings to sender using RTCP, verification of ECN functionality end-to-end, and how to initiate ECN usage. The initiation process will have some dependencies on the signalling mechanism used to establish the RTP session, a specification for mechanisms using SDP is included.

ECN is getting attention as a method to minimise the impact of congestion on real-time multimedia traffic. This as packet loss can be avoided if transmission rate adjustments are quick enough. Including congestion in wireless access networks when radio resources and coverage is insufficient to maintain the current media rates. One key benefit with ECN is it is a lightweight mechanism to allow for each node along the transmission path to set a congestion notification in the IP header, thereby letting the endpoints know of the congested situation.

The introduction of ECN into the Internet requires changes to both the network and transport layers. At the network layer, IP has to be updated to allow routers to mark packets, rather than discarding them in times of congestion [RFC3168]. In addition, transport protocols have to be modified to inform that sender that ECN marked packets are being received, so it can respond to the congestion. TCP [RFC3168], SCTP [RFC4960] and DCCP [RFC4340] have been updated to support ECN, but to date there is no specification how UDP-based transports, such as RTP [RFC3550], can be used with ECN.

The remainder of this memo is structured as follows. We start by describing the conventions, definitions and acronyms used in this memo in Section 2, and the design rationale and applicability in Section 3. The means by which ECN is used with RTP over UDP is defined in Section 4, along with RTCP extensions for ECN feedback in Section 5. In Section 6 we discuss how RTCP ECN feedback is handled in RTP translators. Section 7 discusses some implementation considerations, Section 8 lists IANA considerations, and Section 9 discusses the security considerations.

2. Conventions, Definitions and Acronyms

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 RFC 2119 [RFC2119].

Abbreviations

ECN: Explicit Congestion Notification

ECT: ECN Capable Transport

ECN-CE: ECN Congestion Experienced

not-ECT: Not ECN Capable Transport

3. Discussion, Requirements, and Design Rationale

ECN has been specified for use with TCP [RFC3168], SCTP [RFC4960], and DCCP [RFC4340] transports. These are all unicast protocols which negotiate the use of ECN during the initial connection establishment handshake (supporting incremental deployment, and checking if ECN marked packets pass all middleboxes on the path). ECN Congestion Experienced (ECN-CE) marks are immediately echoed back to the sender by the receiving end-point using an additional bit in feedback messages, and the sender then interprets the mark as equivalent to a packet loss for congestion control purposes.

If RTP is run over TCP, SCTP, or DCCP, it can use the native ECN support provided by those protocols. This memo does not concern itself further with these use cases. However, RTP is more commonly run over UDP. This combination does not currently support ECN, and we observe that it has significant differences from the other transport protocols for which ECN has been specified. These include:

Signalling: RTP relies on separate signalling protocols to negotiate parameters before a session can be created, and doesn't include an in-band handshake or negotiation at session set-up time (i.e. there is no equivalent to the TCP three-way handshake in RTP).

Feedback: RTP does not explicitly acknowledge receipt of datagrams. Instead, the RTP Control Protocol (RTCP) provides reception quality feedback, and other back channel communication, for RTP sessions. The feedback interval is generally on the order of seconds, rather than once per network RTT (although the RTP/AVPF profile [RFC4585] allows more rapid feedback in some cases).

Congestion Response: While it is possible to adapt the transmission of many audio/visual streams in response to network congestion, and such adaptation is required by [RFC3550], the dynamics of the congestion response may be quite different to those of TCP or other transport protocols.

Middleboxes: The RTP framework explicitly supports the concept of mixers and translators, which are middleboxes that are involved in media transport functions.

Multicast: RTP is explicitly a group communication protocol, and was designed from the start to support IP multicast (primarily ASM, although a recent extension supports SSM with unicast feedback).

These differences will significantly alter the shape of ECN support in RTP-over-UDP compared to ECN support in TCP, SCTP, and DCCP, but do not invalidate the need for ECN support. Indeed, in many ways, ECN support is more important for RTP sessions, since the impact of packet loss in real-time audio-visual media flows is highly visible to users. Effective ECN support for RTP flows running over UDP will allow real-time audio-visual applications to respond to the onset of congestion before routers are forced to drop packets, allowing those applications to control how they reduce their transmission rate, and hence media quality, rather than responding to, and trying to conceal the effects of, unpredictable packet loss. Furthermore, widespread deployment for ECN and active queue management in routers, should it occur, can potentially reduce unnecessary queueing delays in routers, lowering the round-trip time and benefiting interactive applications of RTP, such a voice telephony.

3.1. Requirements

Considering ECN and these protocols one can create a set of requirements that must be satisfied to at least some degree if ECN is used by an other protocol (such as RTP over UDP)

- REQ 1: A mechanism to negotiate and initiate the usage of ECN for RTP/UDP/IP sessions is required
- REQ 2: A mechanism to feedback the reception of any packets that are ECN-CE marked to the packet sender is required
- REQ 3: Provide mechanism to minimise the possibility for cheating is preferable
- REQ 4: Some detection and fallback mechanism is needed in case an intermediate node clears ECT or drops packets with ECT set to avoid loss of communication due to the attempted usage of ECN.
- REQ 5: Negotiation of ECN should not significantly increase the time taken to negotiate and set-up the RTP session (an extra RTT before the media can flow is unlikely to be acceptable).

- REQ 6: Negotiation of ECN should not cause clipping at the start of a session.

The following sections describes how these requirements can be meet for RTP over UDP.

3.2. Applicability

The use of ECN with RTP over UDP is dependent on negotiation of ECN capability between the sender and receiver(s), and validation of ECN support in all elements of the network path(s) traversed. RTP is used in a heterogeneous range of network environments and topologies, with various different signalling protocols, all of which need to be verified to support ECN before it can be used.

The usage of ECN is further dependent on a capability of the RTP media flow to react to congestion signalled by ECN marked packets. Depending on the application, media codec, and network topology, this adaptation can occur at the sender by changing the media encoding, at the receiver by changing the subscription to a layered encoding, or in a transcoding middlebox. RFC 5117 identifies seven topologies in which RTP sessions may be configured, and which may affect the ability to use ECN:

Topo-Point-to-Point: This is a standard unicast flow. ECN may be used with RTP in this topology in an analogous manner to its use with other unicast transport protocols, with RTCP conveying ECN feedback messages.

Topo-Multicast: This is either an any source multicast (ASM) group with potentially several active senders and multicast RTCP feedback, or a source specific multicast (SSM) group with a single sender and unicast RTCP feedback from receivers. RTCP is designed to scale to large group sizes while avoiding feedback implosion (see Section 6.2 of [RFC3550], [RFC4585], and [I-D.ietf-avt-rtcpssm]), and can be used by a sender to determine if all its receivers, and the network paths to those receivers, support ECN (see Section 4.2). It is somewhat more difficult to determine if all network paths from all senders to all receivers support ECN. Accordingly, we allow ECN to be used by an RTP sender using multicast UDP provided the sender has verified that the paths to all its known receivers support ECN, and irrespective of whether the paths from other senders to their receivers support ECN. Note that group membership may change during the lifetime of a multicast RTP session, potentially introducing new receivers that are not ECN capable. Senders MUST use the mechanisms described in Section 4.4 to monitor that all receivers continue to support ECN, and MUST fallback to non-ECN use if they do not.

Topo-Translator: An RTP translator is an RTP-level middlebox that is invisible to the other participants in the RTP session (although it is usually visible in the associated signalling session). There are two types of RTP translator: those do not modify the media stream, and are concerned with transport parameters, for example a multicast to unicast gateway; and those that do modify the media stream, for example transcoding between different media codecs. A single RTP session traverses the translator, and the translator must rewrite RTCP messages passing through it to match the changes it makes to the RTP data packets. A legacy, ECN-unaware, RTP translator is expected to ignore the ECN bits on received packets, and zero out the ECN bits when sending packets, so causing ECN negotiation on the path containing the translator to fail (any new RTP translator that does not wish to support ECN may do similarly). An ECN aware RTP translator may act in one of three ways:

* If the translator does not modify the media stream, it should copy the ECN bits unchanged from the incoming to the outgoing datagrams, unless it is overloaded and experiencing congestion, in which case it may mark the outgoing datagrams with an ECN-CE mark. Such a translator passes RTCP feedback unchanged.

* If the translator modifies the media stream to combine or split RTP packets, but does not otherwise transcode the media, it must manage the ECN bits in a way analogous to that described in Section 5.3 of [RFC3168]: if an ECN marked packet is split into two, then both the outgoing packets must be ECN marked identically to the original; if several ECN marked packets are combined into one, the outgoing packet MUST be either ECN-CE marked or dropped if any of the incoming packets are ECN-CE marked, and should have a random ECT mark otherwise. When RTCP ECN feedback packets (Section 5) are received, they must be rewritten to match the modifications made to the media stream (see Section 6.1).

* If the translator is a media transcoder, the output RTP media stream may have radically different characteristics than the input RTP media stream. Each side of the translator must then be considered as a separate transport connection, with its own ECN processing. This requires the translator interpose itself into the ECN negotiation process, effectively splitting the connection into two parts with their own negotiation. Once negotiation has been completed, the translator must generate synthetic RTCP ECN feedback back to the source based on its own reception, and must respond to RTCP ECN feedback received from the receiver(s) (see Section 6.2).

It is recognised that ECN and RTCP processing in an RTP translator that modifies the media stream is non-trivial.

Topo-Mixer: This is an RTP-level middlebox that aggregates multiple RTP streams, mixing them together to generate a new RTP stream. The mixer is visible to the other participants in the RTP session. The RTP flows on each side of the mixer are treated independently for ECN purposes, with the mixer generating its own RTCP ECN feedback, and responding to ECN feedback for data it sends. Since connections are treated independently, it would seem reasonable to allow the transport on one side of the mixer to use ECN, while the transport on the other side of the mixer is not ECN capable.

Topo-Video-switch-MCU: A video switching MCU receives several RTP flows, but forwards only one of those flows onwards to the other participants at a time. The flow that is forwarded changes during the session, often based on voice activity. Since only a subset of the RTP packets generated by a sender are forwarded to the receivers, a video switching MCU can break ECN negotiation (the success of the ECN negotiation depends on the voice activity of the participant at the instant the negotiation takes place - shout if you want ECN). It also breaks congestion feedback and response, since RTP packets are dropped by the MCU depending on voice activity rather than network congestion. This topology is widely used in legacy products, but is NOT RECOMMENDED for new implementations and cannot be used with ECN.

Topo-RTCP-terminating-MCU: In this scenario, each participant runs an RTP point-to-point session between itself and the MCU. Each of these sessions is treated independently for the purposes of ECN and RTCP feedback, potentially with some using ECN and some not.

Topo-Asymmetric: It is theoretically possible to build a middlebox that is a combination of an RTP mixer in one direction and an RTP translator in the other. To quote RFC 5117 "This topology is so problematic and it is so easy to get the RTCP processing wrong, that it is NOT RECOMMENDED to implement this topology."

These topologies may be combined within a single RTP session.

This ECN mechanism is applicable to both sender and receiver controlled congestion algorithms. The mechanism ensures that both senders and receivers will know about ECN-CE markings and any packet losses. Thus the actual decision point for the congestion control is not relevant. This is a great benefit as RTP session can be adapted in a number of ways, such as media sender using TFRC [RFC5348] or other algorithms, or for multicast sessions either a sender based scheme with lowest common rate, or receiver driven mechanism based on

layers to support more heterogeneous paths.

4. Use of ECN with RTP/UDP/IP

The solution for using ECN with RTP consists of a few different pieces that together makes the solution work:

1. Negotiation of the capability to do ECN with RTP/UDP

2. Initiation and initial verification of ECN capable transport

3. Ongoing use of ECN within an RTP session

4. Failure detection, verification and fallback

Before an RTP session can be created, a signalling protocol is used to discover the other participants and negotiate session parameters (see Section 4.1. One of the parameters that can be negotiated is the capability of a participant to support ECN functionality, or otherwise. Note that all participants having the capability of supporting ECN does not necessarily imply that ECN is usable in an RTP session, since there may be middleboxes on the path between the participants which don't support ECN (for example, a firewall that blocks traffic with the ECN bits set).

When a sender joins a session for which all participants claim ECN capability, it must verify if that capability is usable. There are two ways in which this verification may be done (Section 4.2):

- The sender may generate a subset of its RTP data packets with the ECN field set to ECT(0) or ECT(1). Each receiver will then send an RTCP feedback packet indicating the reception of the ECT marked RTP packets. Upon reception of this feedback from each receiver it knows of, the sender can consider ECN functional for its traffic. Each sender does this verification independently of each other. If a new receiver join an existing session it also needs to verify ECN support. If verification fails the sender needs to stop using ECN. As the sender will not know of the receiver prior to it sending RTP or RTCP packets, the sender will wait for the first RTCP packet from the new receiver to determine if that contains ECN feedback or not.

- Alternatively, ECN support can be verified during an initial end-to-end STUN exchange (for example, as part of ICE connection establishment). After having verified connectivity without ECN capability an extra STUN exchange now with the ECN field set to ECT is performed. If successful the paths capability is verified.

Through the use of an extra STUN attribute also support for this solution can be verified through that mechanism.

The first mechanism, using RTP with RTCP feedback, has the advantage of working for all RTP sessions, but the disadvantages of potential clipping if ECN marked RTP packets are discarded by middleboxes, and slow verification of ECN support. The STUN-based mechanism is faster to verify ECN support, but only works in those scenarios supported by end-to-end STUN, such as within an ICE exchange.

Once ECN support has been verified to work for all receivers, a sender marks all its RTP packets as ECT packets, while receivers feedback any CE marks to the sender using RTCP in RTP/AVPF immediate or early feedback mode (see Section 4.3). An RTCP feedback report is sent as soon as possible by the transmission rules for feedback that are in place. This feedback report contains all the CE marks that has been received since the last regular report until the sending of this packet. This is the mechanism to provide the fastest possible feedback to senders about CE marks. On receipt of an RTCP report indicating that CE marked packets were received, the sender must reduce its sending rate as-if packet loss were reported.

RTCP traffic is never ECT marked for the following reason. ECT marked traffic may be dropped if the path is not ECN compliant. As RTCP is used to provide feedback about what has been transmitted and what ECN markings that are received it is important that these are received in cases when ECT marked traffic is not getting through.

The above feedback is not optimised for reliability, therefore an additional procedure is used to ensure more reliable but less timely reporting of the ECN information. The ECN feedback report is also sent in the regular RTCP receiver reports. In this case they include the ECN information covering the last three reporting intervals. That way a loss of ECN-CE report will with high reliability be eventual reported.

There a numerous reasons why the path the RTP packets take from the sender to the receiver may change, e.g. mobility, link failure followed by re-routing around it. Such an event may result in the packet being sent through a node that are ECN non-compliant, thus remarking or dropping packets with ECT set. To prevent this from impacting the application for any longer duration the function of ECN is constantly monitored using the ECN feedback information. By using an ECN nonce over all the received packet that where not ECN-CE marked and reported explicitly the sender can detect if any remarking happens. If ECT marked packets are being dropped that will evident from the RTCP receiver report where the "extended highest sequence number received" field will stop advancing or if the loss is not 100%

the high reported packet loss rates. A sender detecting a possible ECN non-compliance issue can then stop sending ECT marked packets to determine if that allows the packet to be correctly delivered. If the issues can be connected to ECN, then ECN usage is suspended and possibly also re-negotiated.

In the below detailed specification of the behaviour for the different functions the general case will first be discussed. In cases special considerations are needed for middleboxes, multicast usage etc, those will be specially discussed in related subsections.

4.1. Negotiation of ECN Capability

The first stage of ECN negotiation for RTP-over-UDP is to signal support for ECN capability. There are two signalling schemes that may be used, depending on how ECN usage is to be initiated: an SDP extension to indicate that ECN support should be negotiated using RTP and RTCP, and an ICE parameter to indicate that ECN support should be negotiated using STUN as part of an ICE exchange.

An RTP system that supports ECN MUST implement the SDP extension to signal ECN capability as described in Section 4.1.1. It MAY also implement other ECN capability negotiation schemes, such as the ICE extension described in Section 4.1.2.

4.1.1. Signalling ECN Capability using SDP

One new SDP attribute, "a=ecn-capable-rtp", is defined. This is a media level attribute, which MUST NOT be used at the session level. It is not subject to the character set chosen. The aim of this signalling is to indicate the capability of the sender and receivers to support ECN. If all parties have the capability to use ECN then some on-path mechanism must be used to negotiate its use, and to check that all middleboxes on the path support ECN (Section 4.2.1 describes such a mechanism).

When SDP is used with the offer/answer model [RFC3264], the party generating the SDP offer must insert an "a=ecn-capable-rtp" attribute into the media section of the SDP offer of each RTP flow for which it wishes to use ECN. The answering party includes this same attribute in the media sections of the answer if it has the capability, and wishes to, use ECN, or removes it for those flows for which it does not want to use ECN. If the attribute is removed then ECT MUST NOT be used in any direction for that media flow.

When SDP is used in a declarative manner, for example a multicast session using SAP, negotiation of session description parameters is not possible. The "a=ecn-capable-rtp" attribute MAY be added to the

session description to indicate that the sender will use ECN in the RTP session. Receivers MUST NOT join such a session unless they have the capability to understand ECN-marked UDP packets, and can generate RTCP ECN feedback (note that having the capability to use ECN doesn't necessarily imply that the underlying network path between sender and receiver supports ECN).

The "a=ecn-capable-rtp" attribute MAY be used with RTP media sessions using UDP/IP transport. It MUST NOT be used for RTP sessions using TCP, SCTP, or DCCP transport, or for non-RTP sessions.

As described in Section 4.3.3, most RTP sessions using ECN require rapid RTCP ECN feedback, in order that the sender can react to ECN-CE marked packets. If such rapid feedback is required, the use of the Extended RTP Profile for RTCP-Based Feedback (RTP/AVPF) [RFC4585] MUST be signalled.

4.1.2. ICE Parameter to Signal ECN Capability

One new ICE [I-D.ietf-mmusic-ice] option, "rtp+ecn", is defined. This is used with the SDP session level "a=ice-options" attribute in an SDP offer to indicate that the initiator of the ICE exchange has the capability to support ECN for RTP-over-UDP flows (via "a=ice-options: rtp+ecn"). The answering party includes this same attribute at the session level in the SDP answer if it has the capability, and wishes to, use ECN, and removes the attribute if it does not wish to use ECN, or doesn't have the capability to use ECN.

If both sides in the ICE exchange have the capability to use ECN, then they will try to initiate ECN usage using the mechanisms we describe in Section 4.2.2 for any nominated candidate that uses UDP as transport protocol for an RTP session and which also include the "a=ecn-capable-rtp" attribute associated with that media line. They MUST NOT try to initiate ECN usage for RTP sessions using TCP, SCTP, or DCCP transport, or for non-RTP sessions.

As described in Section 4.3.3, most RTP sessions using ECN require rapid RTCP ECN feedback, in order that the sender can react to ECN-CE marked packets. If such rapid feedback is required, the use of the Extended RTP Profile for RTCP-Based Feedback (RTP/AVPF) [RFC4585] MUST be signalled, even when ECN capability negotiation is done through ICE.

4.2. Initiation of ECN Use in an RTP Session

At the start of the RTP session when the first packets with ECT is sent it is important to verify that IP packets with ECN field values of ECT or ECN-CE will reach its destination(s). There is some risk

that the usage of ECN will result in either reset of the ECN field or loss of all packets with ECT or ECN-CE markings. If the path between the sender and the receiver exhibits either of these behaviours one needs to stop using ECN immediately to protect both the network and the application.

The RTP senders and receivers SHALL NOT ECT mark their RTCP traffic during both the initiation and full usage of ECN with RTP. This is to ensure that packet loss due to ECN marking will not effect the RTCP traffic and the necessary feedback information.

An RTP system that supports ECN MUST implement the initiation of ECN using RTP and RTCP described in Section 4.2.1. It MAY also implement other mechanisms to initiate ECN support, for example the STUN-based mechanism described in Section 4.2.2. If support for both mechanisms is signalled, the sender should try ECN negotiation using STUN with ICE first, and if it fails, fallback to negotiation using RTP and RTCP ECN feedback.

No matter how ECN usage is initiated, the sender MUST continually monitor the ability of the network, and all receivers, to support ECN, following the mechanisms described in Section 4.4. This is necessary because path changes or changes in the receiver population may invalidate the ability of the network to support ECN.

4.2.1. Detection of ECT using RTP and RTCP

The ECN initiation phase using RTP and RTCP to detect if the network path supports ECN comprises three stages. Firstly, the RTP sender generates some fraction of its traffic with ECT marks to act a probe for ECN support. Then, on receipt of these ECT-marked packets, the receivers send RTCP ECN feedback packets to inform the sender that their path supports ECN. Finally, the RTP sender makes the decision to use ECN or not, based on whether the paths to all RTP receivers have been verified to support ECN.

Generating ECN Probe Packets: During the ECN initiation phase, an RTP sender SHALL mark a small fraction of its RTP traffic as ECT, while leaving the reminder of the packets unmarked. The reason for only marking some packets is to maintain usable media delivery during the ECN initiation phase in those cases where ECN is not supported by the network path. An RTP sender is RECOMMENDED to send a minimum of two packets with ECT markings per RTCP reporting interval, one with ECT(0) and one with ECT(1), and will continue to send some ECT marked traffic as long as the ECN initiation phase continues. The sender MUST NOT mark all RTP packets as ECT during the ECN initiation phase.

This memo does not mandate which RTP packets are marked with ECT during the ECN initiation phase. An implementation should insert ECT marks in RTP packets in a way that minimises the impact on media quality if those packets are lost. The choice of packets to mark is clearly very media dependent, but the usage of RTP NO-OP payloads [I-D.ietf-avt-rtp-no-op], if supported, would be an appropriate choice. For audio formats, if would make sense for the sender to mark comfort noise packets or similar. For video formats, packets containing P- or B-frames, rather than I-frames, would be an appropriate choice. No matter which RTP packets are marked, those packets MUST NOT be duplicated in transmission, since their RTP sequence number is used to identify packets that are received with ECN markings.

Generating RTCP ECN Feedback: If ECN capability has been negotiated in an RTP session, the participants in the session MUST listen for ECT or ECN-CE marked RTP packets, and generate RTCP ECN feedback packets (Section 5) to mark their receipt. If the use of the Extended RTP Profile for RTCP-Based Feedback (RTP/AVPF) has been negotiated, then an immediate or early (depending on the RTP/AVPF mode) feedback packet SHOULD be generated on receipt of the first ECT or ECN-CE marked packet from a sender that has not previously sent any ECT traffic. If RTP/AVPF has not been negotiated, then the RTCP ECN feedback should be sent in a compound RTCP packet along with the regular RTCP reports. The RTP/AVPF profile SHOULD be negotiated where possible, since it greatly speeds up the ECN initiation phase by ensuring that RTP senders get the earliest possible indication that ECN works.

Determination of ECN Support: RTP is a group communication protocol, where members can join and leave the group at any time. This complicates the ECN initiation phase, since the sender must wait until it believes the group membership has stabilised before it can determine if the paths to all receivers support ECN (group membership changes after the ECN initiation phase has completed are discussed in Section 4.3).

An RTP sender shall consider the group membership to be stable after it has been in the session and sending ECT-marked probe packets for at least three RTCP reporting intervals (i.e. after sending its third regularly scheduled RTCP packet), and when a complete RTCP reporting interval has passed without changes to the group membership. ECN initiation is considered successful when the group membership becomes stable, provided all known participants have sent one or more RTCP ECN feedback packets indicating correct receipt of the ECT-marked RTP packets generated by the sender.

As an optimisation, if an RTP sender is initiating ECN usage towards a unicast address, then it MAY treat the ECN initiation as provisionally successful if it receives a single RTCP ECN feedback report indicating successful receipt of the ECT-marked packets, with no negative indications, from a single RTP receiver. After declaring provisional success, the sender MAY generate ECT-marked packets as described in Section 4.3, provided it continues to monitor the RTCP reports for a period of three RTCP reporting intervals from the time the ECN initiation started, to check if there is more than one other participant in the session. If other participants are detected, the sender MUST fallback to only ECT-marking a small fraction of its RTP packets, while it determines if ECN can be supported following the full procedure described above.

Note: One use case that requires further consideration is a unicast connection with several SSRCs multiplexed onto the same flow (e.g. SVC video using SSRC multiplexing for the layers). It is desirable to be able to rapidly negotiate ECN support for such a session, but the optimisation above fails since the multiple SSRCs make it appear that this is a group communication scenario. It's not sufficient to check that all SSRCs map to a common RTCP CNAME to check if they're actually located on the same device, because there are implementations that use the same CNAME for different parts of a distributed implementation.

ECN initiation is considered to have failed at the instant when any RTP session participant sends an RTCP packet that doesn't contain an RTCP ECN feedback report, but has an RTCP RR with an extended RTP sequence number field that indicates that it should have received multiple (>3) ECT marked RTP packets. This can be due to failure to support the ECN feedback format by the receiver or some middlebox, or the loss of all ECT marked packets. Both indicate a lack of ECN support.

The reception of RTCP ECN feedback packets that indicate greatly increased packet loss rates for ECT marked packets, compared to non-ECT marked packets, is a strong indication of problems with ECN support on the network path. Senders MAY consider such reports as indications that they should not use ECN on the path, even though some ECT-marked packets to reach all receivers.

4.2.2. Detection of ECT using STUN with ICE

This section describes an OPTIONAL method that can be used to avoid media impact and also ensure ECN capable path prior to media transmission. This method is considered in the context where the

session participants are using ICE [I-D.ietf-mmusic-ice] to find working connectivity. We need to use ICE rather than STUN only, as the verification needs to happen from the media sender to the address and port on which the receiver is listening.

To minimise the impact of set-up delay, and to prioritise the fact that one has a working connectivity rather than necessarily finding the best ECN capable network path, this procedure is applied after having performed a successful connectivity check for a candidate, which is nominated for usage. At that point, and provided the chosen candidate is not a relayed address, one performs an additional connectivity check including the here defined STUN attribute "ECT Check" and in an IP/UDP packet that are ECT marked. The STUN server will upon reception of the packet note the received ECN field value and in its response send an IP/UDP/STUN Packet with ECN field set to not-ECT and also include the ECN check STUN attribute.

The STUN ECN check STUN attribute contains one field and a flag. The flag indicate if the echo field contains a valid value or not. The field is the ECN echo field, and when valid contains the two ECN bits from the packet it echoes back. The ECN check STUN attribute is an comprehension optional attribute.

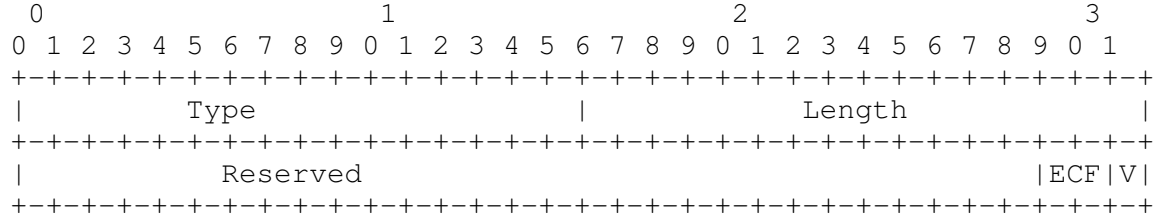

Figure 1: ECN Check Stun Attribute

V: Valid (1 bit) ECN Echo value field is valid when set to 1, and invalid when set 0.

ECF: ECN Echo value field (2 bits) contains the ECN filed value of the STUN packet it echoes back when field is valid. If invalid the content is arbitrary.

Reserved: Reserved bits (29 bits) SHALL be set to 0 and SHALL be ignored on reception.

This attribute MAY be included in any STUN request to request the ECN field to be echoed back. In STUN requests the V bit SHALL be set to 0. A STUN server receiving a request with the ECN Check attribute which understand it SHALL read the ECN field value of the IP/UDP packet the request was received in. Upon forming the response the

server SHALL include the ECN Check attribute setting the V bit to valid and include the read value of the ECN field into the ECF field.

4.3. Ongoing Use of ECN Within an RTP Session

Once ECN usage has been successfully initiated for an RTP sender, that sender begins actively sending ECT-marked RTP data packets, and its receivers begin sending ECN feedback via RTCP packets. This section describes procedures for sending ECT-marked data, providing ECN feedback via RTCP, responding to ECN feedback, and detecting failures and misbehaving receivers.

4.3.1. Transmission of ECT-marked RTP Packets

After a sender has successfully initiated ECN usage, it SHOULD mark all the RTP data packets it sends as ECT. The choice between ECT(0) and ECT(1) MUST be made randomly for each packet, and the sender MUST calculate and record the ECN-nonce sum for outgoing packets [RFC3540] to allow the use of the ECN-nonce to detect receiver misbehaviour (see Section 4.4). Guidelines on the random choice of ECT values are provided in Section 8 of [RFC3540].

The sender SHALL NOT include ECT marks on outgoing RTCP packets, and SHOULD NOT include ECT marks on any outgoing control messages (e.g. STUN [RFC5389] packets, DTLS [RFC4347] handshake packets, or ZRTP [I-D.zimmermann-avt-zrtp] control packets, that are multiplexed on the same UDP port).

4.3.2. Reporting ECN Feedback via RTCP

An RTP receiver that receives a packet with an ECN-CE mark, or that detects a packet loss, MUST schedule the transmission of an RTCP ECN feedback packet as soon as possible to report this back to the sender. The feedback RTCP packet sent SHALL consist at least one ECN feedback packet (Section 5) reporting on the packets received since the last regular RTCP report, and SHOULD contain an RTCP SR or RR packet. The RTP/AVPF profile in early or immediate feedback mode SHOULD be used where possible, to reduce the interval before feedback can be sent. To reduce the size of the feedback message, reduced size RTCP [RFC5506] MAY be used if supported by the end-points. Both RTP/AVPF and reduced size RTCP MUST be negotiated in the session set-up signalling before they can be used.

Every time a regular compound RTCP packet is to be transmitted, the RTP receiver MUST include an ECN feedback packet as part of the compound packet. The ECN feedback packet must report on packets received during the last three reporting intervals unless that would cause the compound RTCP packet to exceed the network MTU, in which

case it MAY be reduced to cover only the last or two last reporting intervals. It is important to configure the RTCP bandwidth (e.g. using an SDP "b=" line) such that the bit-rate is sufficient for a usage that includes ECN-CE events. Each RTCP feedback packet will report on the ECN-CE marks received since the last report and the current ECN nonce value.

The multicast feedback implosion problem, that occurs when many receivers simultaneously send feedback to a single sender, must also be considered. The RTP/AVPF transmission rules will limit the amount of feedback that can be sent, avoiding the implosion problem but also delaying feedback by varying degrees from nothing up to a full RTCP reporting interval. As a result, the full extent of a congestion situation may take some time to reach the sender, although some feedback should arrive reasonably timely, allowing the sender to react on a single or a few reports.

An open issue is whether we should employ some form of feedback suppression on ECN-CE feedback for groups? If one can make an assumption that a sender will react on a few ECN-CE marks then suppression could be employed successfully and reduce the RTCP bandwidth usage.

In case a receiver driven congestion control algorithm is to be used and has through signalling been agreed upon, the algorithm MAY specify that the immediate scheduling (and later transmission) of ECN-CE feedback of any received ECN-CE mark is not required and shall not be done. In that case ECN feedback is only sent using regular RTCP reports for verification purpose and in response to the initiation process of any new media senders as specified in Section 4.2.1.

4.3.3. Response to Congestion Notifications

When RTP packets are received with ECN-CE marks, the sender and/or receivers MUST react with congestion control as-if those packets had been lost. Depending on the media format, type of session, and RTP topology used, there are several different types of congestion control that can be used.

Sender-Driven Congestion Control: The sender may be responsible for adapting the transmitted bit-rate in response to RTCP ECN feedback. When the sender receives the ECN feedback data it feeds this information into its congestion control or bit-rate adaptation mechanism so that it can react on it as if it was packet losses that was reported. The congestion control algorithm to be used is not specified here, although TFRC [RFC5348] is one example that might be used.

Receiver-Driven Congestion Control: If receiver driven congestion control mechanism is used, the receiver can react to the ECN-CE marks without contacting the sender. This may allow faster response than sender-driven congestion control in some circumstances. Receiver-driven congestion control is usually implemented by providing the content in a layered way, with each layer providing improved media quality but also increased bandwidth usage. The receiver locally monitors the ECN-CE marks on received packet to check if it experiences congestion at the current number of layers. If congestion is experienced, the receiver drops one layer, so reducing the resource consumption on the path towards itself. For example, if a layered media encoding scheme such as H.264 SVC is used, the receiver may change its layer subscription, and so reduce the bit rate it receives. The receiver MUST still send RTCP ECN feedback to the sender, even if it can adapt without contact with the sender, so that the sender can determine if ECN is supported on the network path. The timeliness of RTCP feedback is less of a concern with receiver driven congestion control, and regular RTCP reporting of ECN feedback is sufficient (without using RTP/AVPF immediate or early feedback).

Responding to congestion indication in the case of multicast traffic is a more complex problem than for unicast traffic. The fundamental problem is diverse paths, i.e. when different receivers don't see the same path, and thus have different bottlenecks, so the receivers may get ECN-CE marked packets due to congestion in different points in the network. This is problematic for sender driven congestion control, since when receivers are heterogeneous in regards to capacity the sender is limited to transmitting at the rate the slowest receiver can support. This often becomes a significant limitation as group size grows. Also, as group size increases the frequency of reports from each receiver decreases, which further reduces the responsiveness of the mechanism. Receiver-driven congestion control has the advantage that each receiver can choose the appropriate rate for its network path, rather than all having to settle for the lowest common rate.

Note: There are many additional references that may be cited here. If this document is accepted as an AVT work item, some discussion of the appropriate amount of detail to include here would be worthwhile.

We note that ECN support is not a silver bullet to improving performance. The use of ECN gives the change to respond to congestion before packets are dropped in the network, improving the user experience by allowing the RTP application to control how the quality is reduced. An application which ignores ECN congestion

experienced feedback is not immune to congestion: the network will eventually begin to discard packets if traffic doesn't respond. It is in the best interest of an application to respond to ECN congestion feedback promptly, to avoid packet loss.

4.4. Detecting Failures and Receiver Misbehaviour

ECN-nonce is defined in RFC3540 as a means to ensure that a TCP clients does not mask ECN-CE marks, this assumes that the sending endpoint (server) acts on behalf of the network.

The assumption about the senders acting on the behalf of the network may be reduced due to the nature of peer-to-peer usage. Still a large part of RTP senders are infrastructure devices that do have an interest in protecting both service quality and the network. In addition as real-time media commonly is more sensitive to increased delay and packet loss it will be in both media sender and receivers interest to minimise the number and duration of any congestion events as it will affect media quality.

In addition ECN with RTP can suffer from path changes resulting in that a non ECN compliant node becomes part of the path. That node may perform either of two actions that has effect on the ECN and application functionality. The gravest is if the node drops packets with any ECN field values other than 00b. This can be detected by the receiver when it receives a RTCP SR packet indicating that a number of packets has not been received. The sender may also detect it based on the receivers RTCP RR packet where the extended sequence number is not advanced due to the failure to receive packets. If the packet loss is less than 100% then packet loss reporting in either the ECN feedback message or RTCP RR will indicate the situation. The other action is to remark a packet from ECT to not-ECT. That has less dire results, however, it should be detected so that ECN usage can be suspended to prevent misusing the network.

ECN nonce is used as part of this solution primarily to detect non-compliant nodes on the path. Due to its definition it will also detect receivers attempting to cheat. We can note that it appears quite counter productive for a receiver to attempt to cheat as it most likely will have negative impact on its media quality.

The ECN nonce mechanism used is not exactly the same as in RFC 3540 due to the desire to detect also re-markings of ECT to not-ECT. Thus the nonce is the 2-bit XOR sum of the previous packets Nonce value and the ECN field. The initial value for the Nonce is 00b.

Thus packet losses and ECN-nonce failures are possible indication of issues with using ECN over the path. The next section defines both

sender and receiver reactions to these cases.

4.4.1. Fallback mechanisms

Upon the detection of a potential failure both the sender and the receiver can react to mitigate the situation.

A Receiver that detects a packet loss burst MAY schedule an early feedback packet to report this to the sender that includes at least the RTCP RR and the ECN feedback message. Thus speeding up the detection at the sender of the losses and thus triggering sender side mitigation.

A Sender that detects high packet loss rates for its RTP packet flow while sending them marked as ECT, SHOULD immediately remark them as not-ECT to determine if the losses potentially are due to the ECT markings. If the losses disappear with the remarking, the RTP sender should go back to initiation procedures to attempt to verify the apparent loss of ECN capability of the used path. If a re-initiation fails then the two possible actions exist:

1. Periodically retry the ECN initiation to detect if a path change occurs to a path that are ECN capable.

2. Renegotiating the session to disable ECN support. A choice that is suitable if the impact of ECT probing on the media quality are noticeable. If multiple initiations has been successful but the following full usage of ECN has resulted in the fallback procedures then disabling of the ECN support is RECOMMENDED.

We foresee the possibility of flapping ECN capability due to several reasons:

- Video switching MCU or similar middleboxes that selects to deliver media from the sender only intermittently.

- Load balancing devices may in worst case result in that some packets take a different network path then the others.

- Mobility solutions that switches underlying network path in a transparent way for the sender or receiver.

- Membership changes in a multicast group.

5. RTCP Extension for ECN feedback

One AVPF NACK Transport feedback format with the following functionality is defined:

- ECN Nonce
- Explicit Sequence numbers for ECN-CE marked packets
- Explicit Sequence numbers for lost packets

The usage of this feedback format called "ECN feedback format" includes in addition to progressive reporting of ECN-CE marking using Immediate or early feedback also Initiation and verification procedures.

The RTCP packet starts with the common header defined by AVPF [RFC4585] which is reproduced here for the readers information:

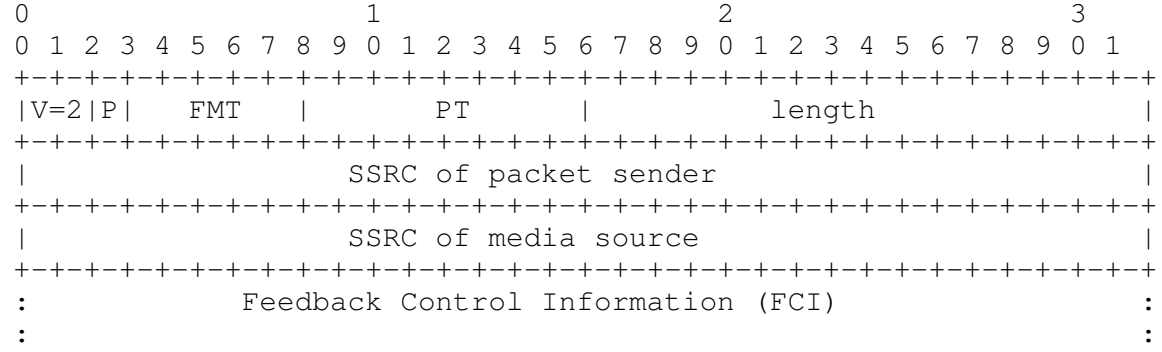

Figure 2: AVPF Feedback common header

From Figure 2 it can be determined the identity of the feedback provider and for which RTP packet sender it applies. Below is the feedback information format defined that is inserted as FCI for this particular feedback messages that is identified with an FMT value=TBA1.

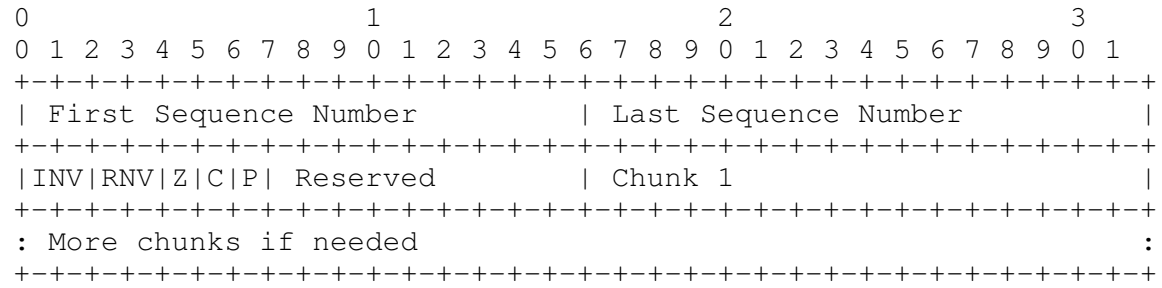

Figure 3: ECN Feedback Format

The FCI information for the ECN Feedback format (Figure 3) are the following:

First Sequence Number: The first RTP sequence number included in the ECN nonce and base sequence number for the run length encoding.

Last Sequence Number The last RTP sequence number included in the ECN nonce and the run length encoding.

INV: Initial Nonce Value. Which is the value of Nonce prior to the XOR addition of the ECN field value for the packet with RTP sequence number of "First Sequence Number". This to allow running calculations and only need to save nonce values at reporting boundaries.

RNV: Resulting Nonce Value. The Nonce sum value resulting after having XOR the ECN field value for all packets received and not ECN-CE marked with the INV value.

Z: ECN Non-capable transport value seen. If set to 1, at least one packet within the feedback interval has had its ECN value set to 00b (Not-ECT). If set to 0, no packets within the reporting interval has its ECN field value set to Not-ECT.

C: ECN-CE value(s) part of the feedback interval. If set to 1, at least one packet within the feedback interval was ECN-CE marked, the sequence numbers of the packets are explicitly encoded using chunks. If set to 0, no packets within the reporting interval had their ECN value set to ECN-CE and no chunks are included.

P: Packet loss part of the feedback interval. If set to 1, at least one packet within the feedback interval was lost in transit, the sequence numbers of the packets are explicitly encoded using chunks. If set to 0, no packets within the reporting interval was lost and no chunks are included.

Each FCI reports on a single source. Multiple sources can be reported by including multiple RTCP feedback messages in an compound RTCP packet. The AVPF common header indicates both the sender of the feedback message and on which stream it relates to.

Both the ECN-CE and packet loss information is structured as bit vector where the first bit represents the RTP packet with the sequence number equal to the First Sequence number. The bit-vector will contain values representing all packets up to and including the one in the "Last Sequence Number" field. The chunk mechanism used to represent the bit-vector in an efficient way may appear longer upon reception if an explicit bit-vector is used as the last chunk. Bit-

values representing packets with higher sequence number (modulo 16) than "Last Sequence Number" are not valid and SHALL be ignored.

The RTP sequence number can easily wrap and that needs to be considered when handling them. The report SHALL NOT report on more than 32768 consecutive packets. The last sequence number is the extended sequence number that is equal too or smaller (less than 65535 packets) than the value present in the Receiver Reports "extended highest sequence number received" field. The "first sequence number" value is thus is as an extended sequence number smaller than the "last sequence number". If there is a wrap between the first sequence number and the last, i.e. First sequence number > Last sequence number (seen as 16-bit unsigned integers), then the wrap needs to included in the calculation.

The ECN-CE bit-vector uses values of 1 to represent that the corresponding packet was marked as ECN-CE, all other ECN values are represented as a 0. The packet loss bit vector uses value of 1 to represent that the corresponding packet was received and a value of 0 to represent loss.

The produced bit-vectors are encoded using chunks. The chunks are any of the three types defined in [RFC3611], Run Length Chunk (Section 4.1.1 of [RFC3611]), Bit Vector Chunk (Section 4.1.2 of [RFC3611]), or Terminating Null Chunk (Section 4.1.3 of [RFC3611]). In the chunk part of the FCI at least one chunk MUST be included to achieve 32-bit word alignment. The C and P bits are used to indicate the inclusion of two different information reports in the feedback message. When both C and P are sent, the chunks reporting if ECN-CE was set SHALL be sent first, followed by one Terminating Null chunk followed by the chunks reporting on which packets where lost, possibly followed by one terminating null chunk to achieve 32-bit word alignment. If only one of the C and P bits are set the chunks reports on only that information, the last chunk MAY be a Terminating Null chunk if necessary to achieve 32-bit word alignment. If none of the C and P bits are set, only a single Terminating Null Chunk is included.

(tbd: We also need to register a regular RTCP packet format containing the same information as the AVPF NACK feedback format, so that it can be used with in regular compound RTCP packets.)

6. Processing RTCP ECN Feedback in RTP Translators and Mixers

RTP translators and mixers that support ECN feedback are required to process, and potentially modify or generate, RTCP packets for the translated and/or mixed streams.

6.1. Fragmentation and Reassembly in Translators

An RTP translator may fragment or reassemble RTP data packets without changing the media encoding. An example of this might be to combine packets of a voice-over-IP stream coded with one 20ms frame per RTP packet into new RTP packets with two 20ms frames per packet, thereby reducing the header overheads and so stream bandwidth, at the expense of an increase in latency. If multiple data packets are re-encoded into one, or vice versa, the RTP translator MUST assign new sequence numbers to the outgoing packets. Losses in the incoming RTP packet stream may induce corresponding gaps in the outgoing RTP sequence numbers. An RTP translator MUST also rewrite RTCP packets to make the corresponding changes to their sequence numbers. This section describes how that rewriting is to be done for RTCP ECN feedback packets. Section 7.2 of [RFC3550] describes general procedures for other RTCP packet types.

(tbd: complete this section)

6.2. Generating RTCP ECN Feedback in Translators

An RTP translator that acts as a media transcoder cannot directly forward RTCP packets corresponding to the transcoded stream, since those packets will relate to the non-transcoded stream, and will not be useful in relation to the transcoded RTP flow. Such a transcoder will need to interpose itself into the RTCP flow, acting as a proxy for the receiver to generate RTCP feedback in the direction of the sender relating to the pre-transcoded stream, and acting in place of the sender to generate RTCP relating to the transcoded stream, to be sent towards the receiver. This section describes how this proxying is to be done for RTCP ECN feedback packets. Section 7.2 of [RFC3550] describes general procedures for other RTCP packet types.

(tbd: complete this section)

6.3. Generating RTCP ECN Feedback in Mixers

An RTP mixer terminates one-or-more RTP flows, combines them into a single outgoing media stream, and transmits that new stream as a separate RTP flow. An ECN-aware RTP mixer must send RTCP reports and provide ECN feedback for the RTP flows it terminates, and must generate RTCP reports for the RTP flow it originates, and add ECT marks to the outgoing packets. This section describes how RTCP is processed in RTP mixers, and how that interacts with ECN feedback.

(tbd: complete this section)

7. Implementation considerations

To allow the use of ECN with RTP over UDP, the RTP implementation must be able to set the ECT bits in outgoing UDP datagrams, and must be able to read the value of the ECT bits on received UDP datagrams. The standard Berkeley sockets API predates the specification of ECN, and does not provide the functionality which is required for this mechanism to be used with UDP flows, making this specification difficult to implement portably.

8. IANA Considerations

Note to RFC Editor: please replace "RFC XXXX" below with the RFC number of this memo, and remove this note.

8.1. SDP Attribute Registration

Following the guidelines in [RFC4566], the IANA is requested to register one new SDP attribute:

- Contact name, email address and telephone number: Authors of RFCXXXX
- Attribute-name: ecn-capable-rtp
- Type of attribute: media-level
- Subject to charset: no

This attribute defines the ability to negotiate the use of ECT (ECN capable transport). This attribute should be put in the SDP offer if the offering party wishes to receive an ECT flow. The answering party should include the attribute in the answer if it wish to receive an ECT flow. If the answerer does not include the attribute then ECT MUST be disabled in both directions.

8.2. AVPF Transport Feedback Message

A new RTCP Transport feedback message needs a FMT code point assigned. ...

8.3. STUN attribute

A new STUN attribute in the Comprehension-optional range needs to be assigned...

8.4. ICE Option

A new ICE option "rtp+ecn" is registered in the non-existing registry which needs to be created.

9. Security Considerations

The usage of ECN with RTP over UDP as specified in this document has the following known security issues that needs to be considered.

External threats to the RTP and RTCP traffic:

Denial of Service affecting RTCP: For an attacker that can modify the traffic between the media sender and a receiver can achieve either of two things. 1. Report a lot of packets as being Congestion Experience marked, thus forcing the sender into a congestion response. 2. Ensure that the sender disable the usage of ECN by reporting failures to receive ECN by setting the Z bit or changing the ECN nonce field. Both Issues, can also be accomplished by injecting false RTCP packets to the media sender. Reporting a lot of CE marked traffic is likely the more efficient denial of service tool as that may likely force the application to use lowest possible bit-rates. The prevention against an external threat is to integrity protect the RTCP feedback information and authenticate the sender of it.

Information leakage: The ECN feedback mechanism exposes the receivers perceived packet loss, what packets it considers to be ECN-CE marked and its calculation of the ECN-none. This is mostly not considered sensitive information. If considered sensitive the RTCP feedback shall be encrypted.

Changing the ECN bits An on-path attacker that see the RTP packet flow from sender to receiver and who has the capability to change the packets can rewrite ECT into ECN-CE thus forcing the sender or receiver to take congestion control response. This denial of service against the media quality in the RTP session is impossible for en end-point to protect itself against. Only network infrastructure nodes can detect this illicit remarking. It will be mitigated by turning off ECN, however, if the attacker can modify its response to drop packets the same vulnerability exist.

Denial of Service affecting the session set-up signalling: If an attacker can modify the session signalling it can prevent the usage of ECN by removing the signalling attributes used to indicate that the initiator is capable and willing to use ECN with RTP/UDP. This attack can be prevented by authentication and

integrity protection of the signalling. We do note that any attacker that can modify the signalling has more interesting attacks they can perform than prevent the usage of ECN, like inserting itself as a middleman in the media flows enabling wire-tapping also for an off-path attacker.

The following are threats that exist from misbehaving senders or receivers:

Receivers cheating A receiver may attempt to cheat and fail to report reception of ECN-CE marked packets. The benefit for a receiver cheating in its reporting would be to get an unfair bit-rate share across the resource bottleneck. It is far from certain that a receiver would be able to get a significant larger share of the resources. That assumes a high enough level of aggregation that there are flows to acquire shares from. The risk of cheating is that failure to react to congestion results in packet loss and increased path delay. To mitigate the risk of cheating receivers the solution include ECN-Nonce that makes it probabilistically unlikely that a receiver can cheat for more than a few packets before being found out. See [RFC3168] and [RFC3540] for more discussion.

Receivers misbehaving: A receiver may prevent the usage of ECN in an RTP session by reporting itself as non ECN capable or simple provide invalid ECN-nonce values. Thus forcing the sender to turn off usage of ECN. In a point-to-point scenario there is little incentive to do this as it will only affect the receiver. Thus failing to utilise an optimisation. For multi-party session there exist some motivation why a receiver would misbehave as it can prevent also the other receivers from using ECN. As an insider into the session it is difficult to determine if a receiver is misbehaving or simply incapable, making it basically impossible in the incremental deployment phase of ECN for RTP usage to determine this. If additional information about the receivers and the network is known it might be possible to deduce that a receiver is misbehaving. If it can be determined that a receiver is misbehaving, the only response is to exclude it from the RTP session and ensure that is doesn't any longer have any valid security context to affect the session.

Misbehaving Senders: The enabling of ECN gives the media packets a higher degree of probability to reach the receiver compared to not-ECT marked ones. However, this is no magic bullet and failure to react to congestion will most likely only slightly delay a buffer under-run, in which its session also will experience packet loss and increased delay. There are some chance that the media senders traffic will push other traffic out of the way without

being effected to negatively. However, we do note that a media sender still needs to implement congestion control functions to prevent the media from being badly affected by congestion events. Thus the misbehaving sender is getting a unfair share. This can only be detected and potentially prevented by network monitoring and administrative entities. See Section 7 of [RFC3168] for more discussion of this issue.

ECN as covert channel: As the ECN fields two bits can be set to two different values for ECT, it is possible to use ECN as a covert channel with a possible bit-rate of one or two bits per packet. For more discussion of this issue please see [I-D.ietf-tsvwg-ecn-tunnel].

We note that the end-point security functions needs to prevent an external attacker from affecting the solution easily are source authentication and integrity protection. To prevent what information leakage there can be from the feedback encryption of the RTCP is also needed. For RTP there exist multiple solutions possible depending on the application context. Secure RTP (SRTP) [RFC3711] does satisfy the requirement to protect this mechanism despite only providing authentication if a entity is within the security context or not. IPsec [RFC4301] and DTLS [RFC4347] can also provide the necessary security functions.

The signalling protocols used to initiate an RTP session also needs to be source authenticated and integrity protected to prevent an external attacker from modifying any signalling. Here an appropriate mechanism to protect the used signalling needs to be used. For SIP/SDP ideally S/MIME [RFC3851] would be used. However, with the limited deployment a minimal mitigation strategy is to require use of SIPS (SIP over TLS) [RFC3261] [I-D.ietf-sip-sips] to at least accomplish hop-by-hop protection.

We do note that certain mitigation methods will require network functions.

10. References

10.1. Normative References

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

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, November 2003.

[RFC5348] Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 5348, September 2008.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

10.2. Informative References

[I-D.ietf-avt-rtcpssm] Schooler, E., Ott, J., and J. Chesterfield, "RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback", draft-ietf-avt-rtcpssm-18 (work in progress), March 2009.

[I-D.ietf-avt-rtp-no-op] Andreasen, F., "A No-Op Payload Format for RTP", draft-ietf-avt-rtp-no-op-04 (work in progress), May 2007.

[I-D.ietf-mmusic-ice] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-19 (work in progress), October 2007.

[I-D.ietf-sip-sips] Audet, F., "The use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", draft-ietf-sip-sips-09 (work in progress), November 2008.

[I-D.ietf-tsvwg-ecn-tunnel] Briscoe, B., "Tunnelling of Explicit Congestion Notification", draft-ietf-tsvwg-ecn-tunnel-02 (work in progress), March 2009.

[I-D.zimmermann-avt-zrtp] Zimmermann, P., Johnston, A., and J. Callas, "ZRTP: Media Path Key Agreement for Secure RTP", draft-zimmermann-avt-zrtp-15 (work in progress),

March 2009.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

[RFC3540] Spring, N., Wetherall, D., and D. Ely, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces", RFC 3540, June 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC3851] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[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.

[RFC4960] Stewart, R., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences", RFC 5506, April 2009.

Authors' Addresses

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 82 87
Email: magnus.westerlund@ericsson.com

Ingemar Johansson
Ericsson
Laboratoriegrand 11
SE-971 28 Lulea
SWEDEN

Phone: +46 73 0783289
Email: ingemar.s.johansson@ericsson.com

Colin Perkins
University of Glasgow
Department of Computing Science
Glasgow G12 8QQ
United Kingdom

Email: csp@csperkins.org