Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 1, 2011

M. Westerlund
I. Johansson
Ericsson
C. Perkins
University of Glasgow
P. O'Hanlon
UCL
K. Carlberg
G11

January 28, 2011

Explicit Congestion Notification (ECN) for RTP over UDP
draft-ietf-avtcore-ecn-for-rtp-00

Abstract

This document specifies how explicit congestion notification (ECN) can be used with RTP/UDP flows that use RTCP as feedback mechanism. It defines one RTCP XR extension for ECN summary, a RTCP transport feedback format for timely reporting of congestion events, and an STUN extension used in the optional initilization method using ICE. Signalling and procedures for negotiation of capabilities and initilization methods are also defined.

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 August 1, 2011.

Copyright Notice

Table of Contents

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 a feedback mechanism. The solution consists of feedback of ECN congestion experienced markings to the 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 signalling mechanisms using SDP is included.

ECN is getting attention as a method to minimise the impact of congestion on real-time multimedia traffic. When ECN is used, the network can signal to applications that congestion is occurring, whether that congestion is due to queuing at a congested link, limited resources and coverage on a radio link, or other reasons.

ECN provides a way for networks to send congestion control signals to a media transport without having to impair the media. Unlike losses, the signals unambiguously indicate congestion to the transport as quickly as feedback delays allow, and without confusing congestion with losses that might have occurred for other reasons such as transmission errors, packet-size errors, routing errors, badly implemented middleboxes, policy violations and so forth.

The introduction of ECN into the Internet requires changes to both the network and transport layers. At the network layer, IP forwarding 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 the 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 use ECN. This is due to the lack of feedback mechanisms directly in UDP. Instead the signaling control protocol on top of UDP needs to provide that feedback, which for RTP is RTCP.

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. Section 4 provides an overview of how ECN is used with RTP over UDP. Then the definition of the RTCP extensions for ECN feedback in Section 5. Then the SDP signalling extensions required are specified Section 6.Then the full details of how ECN is used with RTP over UDP is defined in Section 7. In Section 8 we discuss how RTCP ECN feedback is handled in RTP translators and mixers.

Section 9 discusses some implementation considerations, Section 10 lists IANA considerations, and Section 11 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

The meaning of the term ECN support depends on which entity between the sender and receiver (inclusive) that is considered. We distinguish between:

- ECN-Capable Host: Sender or receiver of media.
- ECN-Capable Transport: ECT = all ends are ECN capable hosts.
- ECN-Capable Packets: Packets are either ECT or CE.
- ECN-Oblivious Relay: Router or middlebox that treats ECN-Capable Packets no differently from Not-ECT.
- ECN-Capable Queue: Supports ECN marking of ECN-Capable Packets.
- ECN-Blocking Middlebox: Discards ECN-Capable Packets.
- ECN-Reverting Middlebox: Changes ECN-Capable Packets to Not-ECT.

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 most 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 [RFC5760]).

Application Awareness: ECN support via TCP, DCCP, and SCTP constrain the awareness and reaction to packet loss within those protocols. By adding support of ECN through RTCP, the application is made aware of packet loss and may choose one or more approaches in response to that loss.

Counting vs Detecting Congestion: TCP and the protocols derived from it are mainly designed to respond the same whether they experience a burst of congestion indications within one RTT or just one. Whereas real-time applications may be concerned with the amount of congestion experienced, whether it is distributed smoothly or in bursts. When feedback of ECN was added to TCP [RFC3168], the receiver was designed to flip the echo congestion experienced (ECE) flag to 1 for a whole RTT then flop it back to zero. Whereas ECN feedback in RTCP will need to report a count of how much congestion has been experienced within an RTCP reporting period, irrespective of round trip times.

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.

ECN support is more important for RTP sessions than, for instance, is the case for TCP. This is because 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 as voice telephony.

3.1. Requirements

Considering ECN, transport protocols supporting ECN, and RTP based applications one can create a set of requirements that must be satisfied to at least some degree if ECN is to used by RTP over UDP.

- REQ 1: A mechanism MUST negotiate and initiate the usage of ECN for RTP/UDP/IP sessions so that an RTP sender will not send packets with ECT in the IP header unless it knows all potential receivers will understand any CE indications they might receive.

- REQ 2: A mechanism MUST feedback the reception of any packets that are ECN-CE marked to the packet sender

- REQ 3: Provided mechanism SHOULD minimise the possibility for cheating

- REQ 4: Some detection and fallback mechanism SHOULD exist to avoid loss of communication due to the attempted usage of ECN in case an intermediate node clears ECT or drops packets that are ECT marked.

- 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 for some use cases).

- REQ 6: Negotiation of ECN SHOULD NOT cause media 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 in various forms and at various nodes. As an example, the sender can change the media encoding, or the receiver can change the subscription to a layered encoding, or either reaction can be accomplished by 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 [RFC3569] with potentially several active senders and multicast RTCP feedback, or a source specific multicast (SSM) group [RFC4607] 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 [RFC5760]), and can be used by a sender to determine if all its receivers, and the network paths to those receivers, support ECN (see Section 7.2). 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 or have a path that doesn't support ECN. Senders must use the mechanisms described in Section 7.4 to monitor that all receivers continue to support ECN, and they need to fallback to non-ECN use if any senders 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 to set the ECN bits to not-ECT 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 so 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. If the outgoing combined packet is not ECN-CE marked, then it MUST be ECT marked if any of the incoming packets were ECT marked. When RTCP ECN feedback packets (Section 5) are received, they must be rewritten to match the modifications

made to the media stream (see Section 8.1).

* 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 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 8.2).

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

Topo-Mixer: A mixer 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, and is also usually visible in the associated signalling 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, if this is desired.

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 may depend 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.

The ECN mechanism defined in this memo 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 the rate of an RTP session can be varied in a number of ways, for example a unicast media sender might use TFRC [RFC5348] or some other algorithm, while a multicast session could use a sender based scheme adapting to the lowest common supported rate, or a receiver driven mechanism using layered coding to support more heterogeneous paths.

To ensure timely feedback of CE marked packets when needed, this mechanism requires support for the RTP/AVPF profile [RFC4585] or any of its derivatives, such as RTP/SAVPF [RFC5124]. The standard RTP/AVP profile [RFC3551] does not allow any early or immediate transmission of RTCP feedback, and has a minimal RTCP interval whose default value (5 seconds) is many times the normal RTT between sender and receiver.

3.3. Interoperability

The interoperability requirements for this specification are that there is at least one common interoperability point for all implementations. Since initialization using RTP and RTCP is the one method that works in all cases, although is not optimal for all usages, it is selected as mandatory to implement this initialisation method. This method requires both the RTCP XR extension and the ECN feedback format, which requires the RTP AVPF profile to ensure timely feedback.

When one considers all the uses of ECN for RTP it is clear that congestion control mechanisms that are receiver driven only (Section 7.3.3) do not require timely feedback of congestion events. If such a congestion control mechanism is combined with an initialization method that also doesn't require timely feedback using RTCP, like the leap of faith or the ICE based method then neither the ECN feedback format nor AVPF is strictly needed. However, we would like to point out that fault detection can be improved by using receiver side detection (Section 7.4.1) and early reporting of such cases using the ECN feedback mechanism.

For interoperability we do mandate the implementation of AVPF, with both RTCP extensions and the necessary signalling to support a common operations mode. This specification will still recommend the usage of AVPF in all cases as negotiation of the common interoperability point requires AVPF, and mixed negotiation of AVP and AVPF depending on other SDP attributes in the same media block are difficult and the fact that fault detection can be improved when using AVPF. The use of the ECN feedback format is also recommended but cases where there is no requirement for timely feedback will be noted. The term "no timely feedback required" will be used to indicate usage that employs this specification in combination with receiver driven congestion control, and initialization methods that do not require timely feedback, i.e. currently leap of faith and ICE based. We also note that any receiver driven congestion control solution that still requires RTCP for signalling of any adaptation information to the sender will still require AVPF.

4. Overview of Use of ECN with RTP/UDP/IP

The solution for using ECN with RTP over UDP/IP consists of four different pieces that together make the solution work:

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

2. Initiation and initial verification of ECN capable transport

3. Ongoing use of ECN within an RTP session

4. Handling of dynamic groups through failure detection, verification and fallback

The solution includes a new SDP attribute (Section 6.1), the definition of new extensions to RTCP (Section 5) and STUN (Section 7.2.2).

Before an RTP session can be created, a signalling protocol is often used to discover the other participants and negotiate session parameters (see Section 7.1). At the minimum a signalling protocol is used to configure RTP session participants through a declarative method. 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 pass ECN-marked packets (for example, a firewall that blocks traffic with the ECN bits set). This document defines the information that needs to be negotiated, and provides a

mapping to SDP for use in both declarative and offer/answer contexts.

When a sender joins a session for which all participants claim ECN capability, it must verify if that capability is usable. There are three ways in which this verification may be done (Section 7.2):

- The sender may generate a (small) 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 joins an existing session it will reveal whether or not it supports ECN when it sends its first RTCP report to each source. If the RTCP report includes ECN information, verification will have succeeded and sources can continue to send ECT packets. If not, verification fails and each sender MUST stop using ECN.

- 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, this time with the ECN field set to ECT(0) or ECT(1), is performed. If successful the path's capability to convey ECN marked packets is verified. A new STUN attribute is defined to convey feedback that the ECT marked STUN request was received (see Section 7.2.2), along with an ICE signalling option (Section 6.4).

- Thirdly, the sender may make a leap of faith that ECN will work. This is only recommended for applications that know they are running in controlled environments where ECN functionality has been verified through other means. In this mode it is assumed that ECN works, and the system reacts to failure indicators if the assumption proved wrong. The use of this method relies on a high confidence that ECN operation will be successful, or an application where failure is not serious. The impact on the network and other users must be considered when making a leap of faith, so there are limitations on when this method is allowed.

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. The third one, leap-of-faith, has the advantage of avoiding additional tests or complexities and enabling ECN usage from the first media packet. The

downside is that if the end-to-end path contains middleboxes that do not pass ECN, the impact on the application can be severe: in the worst case, all media could be lost if a middlebox that discards ECN marked packets is present. A less severe effect, but still requiring reaction, is the presence of a middlebox that re-marks ECT marked packets to non-ECT, possibly marking packets with a CE mark as non-ECT. This can force the network into heavy congestion due to non-responsiveness, and seriously impact media quality.

Once ECN support has been verified (or assumed) to work for all receivers, a sender marks all its RTP packets as ECT packets, while receivers rapidly feedback any CE marks to the sender using RTCP in RTP/AVPF immediate or early feedback mode, unless no timely feedback is required. An RTCP feedback report is sent as soon as possible according to the transmission rules for feedback that are in place. This feedback report indicates the receipt of new CE marks since the last ECN feedback packet, and also counts the total number of CE marked packets through a cumulative sum. This is the mechanism to provide the fastest possible feedback to senders about CE marks. On receipt of a CE marked packet, the system must react to congestion as-if packet loss has been reported. Section 7.3 describes the ongoing use of ECN within an RTP session.

This rapid feedback is not optimised for reliability, therefore an additional procedure, the RTCP ECN summary reports, is used to ensure more reliable, but less timely, reporting of the ECN information. The ECN summary report contains the same information as the ECN feedback format, only packed differently for better efficiency with reports for many sources. It is sent in a compound RTCP packet, along with regular RTCP reception reports. By using cumulative counters for seen CE, ECT, not-ECT, and packet loss the sender can determine what events have happened since the last report, independently of any RTCP packets having been lost.

RTCP traffic MUST NOT be 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.

There are 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 is ECN non-compliant, thus re-marking or dropping packets with ECT set. To prevent this from impacting the application for longer than necessary, the operation of ECN is constantly monitored by all senders. Both the RTCP ECN summary reports and the ECN feedback packets allow the sender to

compare the number of ECT(0), ECT(1), and non-ECT marked packets received with the number that were sent, while also reporting CE marked and lost packets. If these numbers do not agree, it can be inferred that the path does not reliably pass ECN-marked packets (Section 7.4.2 discusses how to interpret the different cases). A sender detecting a possible ECN non-compliance issue should then stop sending ECT marked packets to determine if that allows the packets to be correctly delivered. If the issues can be connected to ECN, then ECN usage is suspended and possibly also re-negotiated.

5. RTCP Extensions for ECN feedback

This documents defines two different RTCP extensions: one RTP/AVPF [RFC4585] transport layer feedback format for urgent ECN information, and one RTCP XR [RFC3611] ECN summary report block type for regular reporting of the ECN marking information. The full definition of these extensions usage as part of the complete solution is laid out in Section 7.

5.1. RTP/AVPF Transport Layer ECN Feedback packet

This RTP/AVPF transport layer feedback format is intended for usage in AVPF early or immediate feedback modes when information needs to urgently reach the sender. Thus its main use is to report on reception of an ECN-CE marked RTP packet so that the sender may perform congestion control, or to speed up the initiation procedures by rapidly reporting that the path can support ECN-marked traffic. The feedback format is also defined with reduced size RTCP [RFC5506] in mind, where RTCP feedback packets may be sent without accompanying Sender or Receiver Reports that would contain the Extended Highest Sequence number and the accumulated number of packet losses. Both are important for ECN to verify functionality and keep track of when CE marking does occur.

The RTP/AVPF transport layer feedback packet starts with the common header defined by the RTP/AVPF profile [RFC4585] which is reproduced here for the reader’s information:

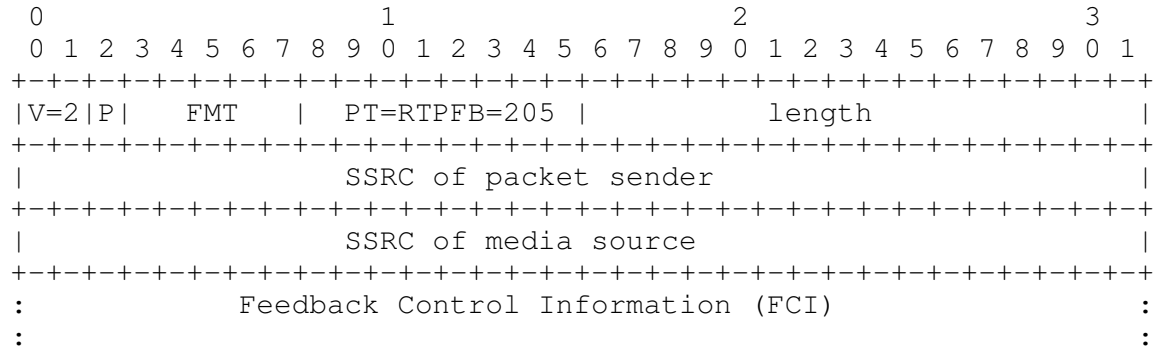


Figure 1: RTP/AVPF Common Packet Format for Feedback Messages

From Figure 1 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].

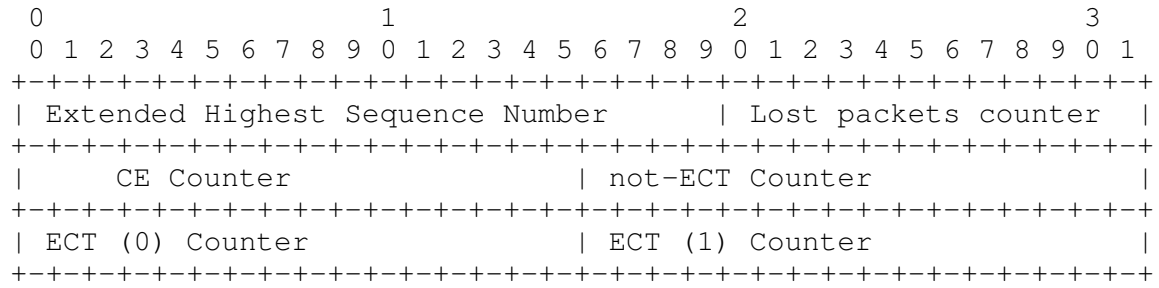


Figure 2: ECN Feedback Format

The FCI information for the ECN Feedback format (Figure 2) are the following:

Extended Highest Sequence Number: The least significant 20-bits from an Extended highest sequence number received value as defined by [RFC3550]. Used to indicate for which packet this report is valid up to.

Lost Packets Counter: The cumulative number of RTP packets that the receiver expected to receive from this SSRC, minus the number of packets it actually received. This is the same as the cumulative number of packets lost defined in Section 6.4.1 of [RFC3550] except represented in 12-bit signed format, compared to 24-bit in RTCP SR or RR packets. As with the equivalent value in RTCP SR or RR packets, note that packets that arrive late are not counted as lost, and the loss may be negative if there are duplicates.

CE Counter: The cumulative number of RTP packets received from this SSRC since the receiver joined the RTP session that were ECN-CE marked. The receiver should keep track of this value using a local representation that is longer than 16-bits, and only include the 16-bits with least significance. In other words, the field will wrap if more than 65535 packets has been received.

ECT(0) Counter: The cumulative number of RTP packets received from this SSRC since the receiver joined the RTP session that had an ECN field value of ECT(0). The receiver should keep track of this value using a local representation that is longer than 16-bits, and only include the 16-bits with least significance. In other words, the field will wrap if more than 65535 packets have been received.

ECT(1) Counter: The cumulative number of RTP packets received from this SSRC since the receiver joined the RTP session that had an ECN field value of ECT(1). The receiver should keep track of this value using a local representation that is longer than 16-bits, and only include the 16-bits with least significance. In other words, the field will wrap if more than 65535 packets have been received.

not-ECT Counter: The cumulative number of RTP packets received from this SSRC since the receiver joined the RTP session that had an ECN field value of not-ECT. The receiver should keep track of this value using a local representation that is longer than 16-bits, and only include the 16-bits with least significance. In other words, the field will wrap if more than 65535 packets have been received.

Each FCI block reports on a single source (SSRC). 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.

The counters SHALL be initiated to 0 for a new receiver. This to enable detection of CE or Packet loss already on the initial report from a specific participant.

The Extended Highest sequence number and packet loss fields are both truncated in comparison to the RTCP SR or RR versions. This is to save bits as the representation is redundant unless reduced size RTCP is used in such a way that only feedback packets are transmitted, with no SR or RR in the compound RTCP packet. Due to that fact regular RTCP reporting will include the longer versions of the fields and there will be less of an issue with wrapping unless the packet rate of the application is so high that the fields will wrap within a

regular RTCP reporting interval. In that case the feedback packet will need to be sent in a compound packet together with the SR or RR packet.

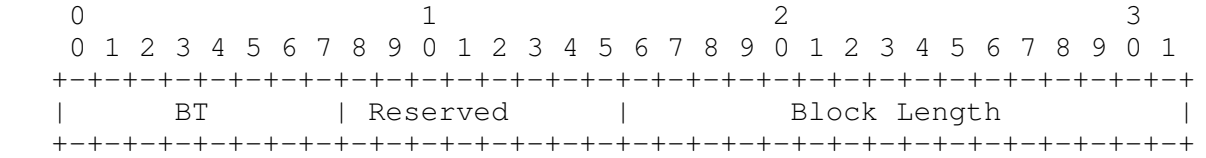
There is an issue with packet duplication in relation to the packet loss counter. If one avoids holding state for which sequence number has been received then the way one can count loss is to count the number of received packets and compare that to the number of packets expected. As a result a packet duplication can hide a packet loss. If a receiver is tracking the sequence numbers actually received and suppresses duplicates it provides for a more reliable packet loss indication. Reordering may also result in that packet loss is reported in one report and then removed in the next.

The CE counter is actually more robust for packet duplication. Adding each received CE marked packet to the counter is not an issue. If one of the clones was CE marked that is still a indication of congestion. Packet duplication has potential impact on the ECN verification. Thus the sum of packets reported may be higher than the number sent. However, most detections are still applicable.

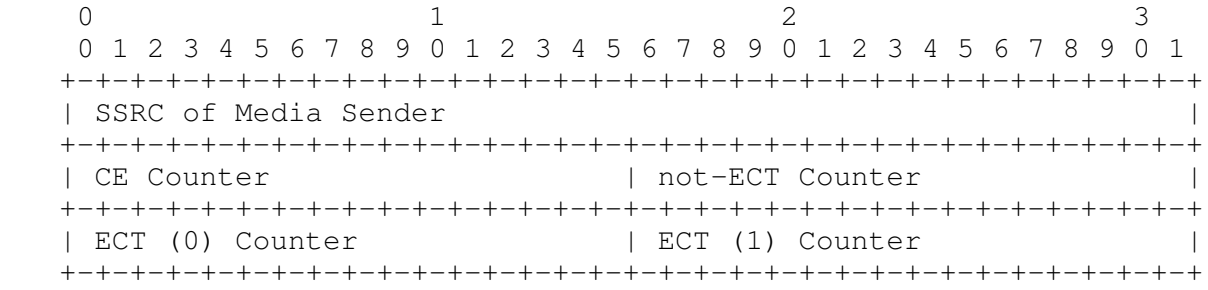
5.2. RTCP XR Report block for ECN summary information

This report block combined with RTCP SR or RR report blocks carries the same information as the ECN Feedback Packet and shall be based on the same underlying information. However, there is a difference in semantics between the feedback format and this XR version. Where the feedback format is intended to report on a CE mark as soon as possible, this extended report is for the regular RTCP report and continuous verification of the ECN functionality end-to-end.

The ECN Summary report block consists of one report block header:



and then followed of one or more of the following report data blocks:



BT: Block Type identifying the ECN summary report block. Value is [TBA2].

Reserved: All bits SHALL be set to 0 on transmission and ignored on reception.

Block Length: The length of the report block. Used to indicate the number of report data blocks present in the ECN summary report. This length will be 3*n, where n is the number of ECN summary report blocks, since blocks are a fixed size.

SSRC of Media Sender: The SSRC identifying the media sender this report is for.

CE Counter: as in Section 5.1.

ECT(0) Counter: as in Section 5.1.

ECT(1) Counter: as in Section 5.1.

not-ECT Counter: as in Section 5.1.

The Extended Highest Sequence number and the packet loss counter for each SSRC is not present in RTCP XR report, in contrast to the feedback version. The reason is that this summary report will rely on the information sent in the Sender Report (SR) or Receiver Report (RR) blocks part of the same RTCP compound packet. The information available in SR or RR are the Extended Highest Sequence number and the accumulated number of packet losses.

All the SSRCs that are present in the SR or RR SHALL also be included in the RTCP XR ECN summary report. In cases where the number of senders are so large that the combination of SR/RR and the ECN summary for all the senders exceed the MTU, then only a subset of the senders SHOULD be included so that the reports for the subset fits within the MTU. The subsets SHOULD be selected round-robin across multiple intervals so that all sources are reported.

6. SDP Signalling Extensions for ECN

This section defines a number of SDP signalling extensions used in the negotiation of the ECN for RTP support when using SDP. This include one SDP attribute "ecn-capable-rtp" that negotiates the actual operation of ECN for RTP. Two SDP signalling parameters are defined to indicate the usage of the RTCP XR ECN summary block and the AVPF feedback format for ECN. One ICE option SDP reprensenation is also defined.

6.1. 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, and to negotiate the method of ECN initiation to be used in the session. The attribute takes a list of initiation methods, ordered in decreasing preference. The defined values for the initiation method are:

rtp: Using RTP and RTCP as defined in Section 7.2.1.

ice: Using STUN within ICE as defined in Section 7.2.2.

leap: Using the leap of faith method as defined in Section 7.2.3.

Further methods may be specified in the future, so unknown methods MUST be ignored upon reception.

In addition, a number of OPTIONAL parameters may be included in the "a=ecn-capable-rtp" attribute as follows:

mode: This parameter signals the endpoint’s capability to set and read ECN marks in UDP packets. An examination of various operating systems has shown that end-system support for ECN marking of UDP packets may be symmetric or asymmetric. By this we mean that some systems may allow end points to set the ECN bits in an outgoing UDP packet but not read them, while others may allow applications to read the ECN bits but not set them. This either/or case may produce an asymmetric support for ECN and thus should be conveyed in the SDP signalling. The "mode=setread" state is the ideal condition where an endpoint can both set and read ECN bits in UDP packets. The "mode=setonly" state indicates that an endpoint can set the ECT bit, but cannot read the ECN bits from received UDP packets to determine if upstream congestion occurred. The "mode=readonly" state indicates that the endpoint can read the ECN bits to determine if congestion has occurred for

incomming packet, but it cannot set the ECT bits in outgoing UDP packets. When the "mode=" parameter is omitted it is assumed that the node has "setread" capabilities. This option can provide for an early indication that ECN cannot be used in a session. This would be case when both the offerer and answerer set the "mode=" parameter to "setonly" or "readonly", or when an RTP sender entity considers offering "readonly".

ect: This parameter makes it possible to express the preferred ECT marking. This is either "random", "0", or "1", with "0" being implied if not specified. The "ect" parameter describes a receiver preference, and is useful in the case where the receiver knows it is behind a link using IP header compression, the efficiency of which would be seriously disrupted if it were to receive packets with randomly chosen ECT marks. It is RECOMMENDED that ECT(0) marking be used.

The ABNF [RFC5234] grammar for the "a=ecn-capable-rtp" attribute is as follows:

```
ecn-attribute  = "a=ecn-capable-rtp:" SP init-list [SP parm-list]
init-list      = init-value *("," init-value)
init-value     = "rtp" / "ice" / "leap" / init-ext
init-ext       = token
parm-list      = parm-value *(";" SP parm-value)
parm-value     = mode / ect / parm-ext
mode           = "mode=" ("setonly" / "setread" / "readonly")
ect            = "ect=" ("0" / "1")
parm-ext       = parm-name "=" parm-value-ext
parm-name      = token
parm-value-ext = token / quoted-string
quoted-string  = DQUOTE *qdtext DQUOTE
qdtext         = %x20-21 / %x23-7E / %x80-FF
                 ; any 8-bit ascii except <">

; external references:
  ; token: from RFC 4566
  ; SP and DQUOTE from RFC 5234
```

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 attribute includes one or more ECN initiation methods in a comma separated list in decreasing order of preference, with any number of optional parameters following. The answering party compares the list of initiation methods in the offer with those it supports in order of preference. If there is a match, and if the receiver wishes to attempt to use ECN in the session, it includes an

"a=ecn-capable-rtp" attribute containing its single preferred choice of initiation method in the media sections of the answer. If there is no matching initiation method capability, or if the receiver does not wish to attempt to use ECN in the session, it does not include an "a=ecn-capable-rtp" attribute in its answer. If the attribute is removed in the answer then ECN MUST NOT be used in any direction for that media flow. If there are initilization methods that are unknown, they MUST be ignored on reception and MUST NOT be included in an answer. The answer may also include optional parameters, as discussed below.

If the "mode=setonly" parameter is present in the "a=ecn-capable-rtp" attribute of the offer and the answering party is also "mode=setonly", then there is no common ECN capability, and the answer MUST NOT include the "a=ecn-capable-rtp" attribute. Otherwise, if the offer is "mode=setonly" then ECN may only be initiated in the direction from the offering party to the answering party.

If the "mode=readonly" parameter is present in the "a=ecn-capable-rtp" attribute of the offer and the answering party is "mode=readonly", then there is no common ECN capability, and the answer MUST NOT include the "a=ecn-capable-rtp" attribute. Otherwise, if the offer is "mode=readonly" then ECN may only be initiated in the direction from the answering party to the offering party.

If the "mode=setread" parameter is present in the "a=ecn-capable-rtp" attribute of the offer and the answering party is "setonly", then ECN may only be initiated in the direction from the answering party to the offering party. If the offering party is "mode=setread" but the answering party is "mode=readonly", then ECN may only be initiated in the direction from the offering party to the answering party. If both offer and answer are "mode=setread", then ECN may be initiated in both directions. Note that "mode=setread" is implied by the absence of a "mode=" parameter in the offer or the answer.

If an RTP session using multicast is negotiated using offer/answer all participants intending to send RTP packets SHALL support setting of ECN packet, and all participants receiving SHALL have the capability to read ECN values on incoming packets. If the participant lack the functionality they SHALL refuse the media streams using multicast.

The "ect=" parameter in the "a=ecn-capable-rtp" attribute is set independently in the offer and the answer. Its value in the offer indicates a preference for the sending behaviour of the answering party, and its value in the answer indicates a sending preference for

the behaviour of the offering party. It will be the senders choice to honour the receivers preference for what to receive or not. In multicast sessions, any sender SHOULD send using the value declared in the ect parameter.

Unknown optional parameters MUST be ignored on reception, and MUST NOT be included in the answer. That way new parameters may be introduced and verified to be supported by the other end-point by having them include it in any answer.

When SDP is used in a declarative manner, for example in a multicast session using the Session Announcement Protocol (SAP, [RFC2974]), 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. The attribute MUST include a single method of initiation. Participants MUST NOT join such a session unless they have the capability to receive ECN-marked UDP packets, implement the method of initiation, 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 mode parameter MAY be included also in declarative usage, to indicate the minimal capability is required by the consumer of the SDP. So for example in a SSM session the participants configured with a particular SDP will all be in a media receive only mode, thus mode=readonly will work as the capability of reporting on the ECN markings in the received is what is required. However, using "mode=readonly" also in ASM sessions is reasonable, unless all senders are required to attempt to use ECN for there outgoing sessions, in which case the mode needs to be set to "setread".

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 7.3.3, RTP sessions using ECN require rapid RTCP ECN feedback, unless timely feedback is not required due to a receiver driven congestion control. To ensure that the sender can react to ECN-CE marked packets timely feedback is usually required. Thus, the use of the Extended RTP Profile for RTCP-Based Feedback (RTP/AVPF) [RFC4585] or other profile that inherits AVPF's signalling rules, MUST be signalled unless timely feedback is not required in case it is RECOMMENDED to be used. The signalling of an AVPF based profile is likely to be required even if the preferred method of initialization and the congestion control does not require timely feedback, as the common interoperable method is likely to be signalled or the improved fault reaction is desired.

6.2. RTCP Feedback SDP Parameter

A new "nack" feedback parameter "ecn" is defined to indicate the usage of the RTCP ECN feedback packet format (Section 5.1). The ABNF [RFC5234] definition of the SDP parameter extension is:

```
rtcp-fb-nack-param  = <See section 4.2 of RFC 4585>
rtcp-fb-nack-param /= ecn-fb-par
ecn-fb-par          = SP "ecn"
```

The offer/answer rules for this SDP feedback parameters are specified in AVPF [RFC4585].

6.3. XR Block SDP Parameter

A new RTCP XR block for ECN summary information is specified, thus the XR block SDP signalling also needs to be extended with a parameter. This is done in the same way as for the other XR blocks. The XR block SDP attribute as defined in Section 5.1 of the RTCP XR specification [RFC3611] is defined to be extendible. As no parameter values are needed for this ECN summary block, this parameter extension consistis of a simple parameter name used to indicate support and intent to use the XR block.

```
xr-format       = <See Section 5.1 of [RFC3611]>
xr-format      /= ecn-summary-par
ecn-summary-par = "ecn-sum"
```

For SDP declarative and offer/answer usage, see the RTCP XR specification[RFC3611].

6.4. ICE Parameter to Signal ECN Capability

One new ICE [RFC5245] 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 also has the capability, and removes the attribute if it does not wish to use ECN, or doesn't have the capability to use ECN. If this initiation method (Section 7.2.2) actually is going to be used, it is explicitly negotiated using the "a=ecn-capable-rtp" attribute.

> Note: This signalling mechanism is not strictly needed as long as the STUN ECN testing capability is used within the context of this document. It may however be useful if the ECN verification capability is used in additional contexts.

7. Use of ECN with RTP/UDP/IP

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

7.1. Negotiation of ECN Capability

The first stage of ECN negotiation for RTP-over-UDP is to signal the capability to use ECN. This includes negotiating if ECN is to be used symmetrically and the method for initial ECT verification. This memo defines the mappings of this information onto SDP for both declarative and offer/answer usage. There is one SDP extension to indicate if ECN support should be used, and the method for initiation (Section 6.1). Further parameters to indicate support for the AVPF ECN feedback format (Section 6.2) and the ECN XR summary report (Section 6.3). In addition an ICE parameter is defined (Section 6.4) to indicate that ECN initiation using STUN is supported as part of an ICE exchange.

An RTP system that supports ECN and uses SDP in the signalling MUST implement the SDP extension to signal ECN capability as described in Section 6.1, the ECN feedback SDP parameter Section 6.2, and the ECN XR SDP parameter Section 6.3. It MAY also implement alternative ECN capability negotiation schemes, such as the ICE extension described in Section 6.4.

The "ecn-capable-rtp" SDP attribute MUST always be used when employing ECN for RTP according to this specification. As the XR ECN summary report is required independently of the initialization method, or congestion control scheme the "rtcp-xr" attribute with the "ecn-sum" parameter MUST also be used. The "rtcp-fb" attribute with the "nack" parameter "ecn" MUST be used whenever the initialization method or a congestion control algorithm requiring timely sender side knowledge of received CE markings.

7.2. Initiation of ECN Use in an RTP Session

Once the sender and the receiver(s) have agreed that they have the capability to use ECN within a session, they may attempt to initiate ECN use.

At the start of the RTP session, when the first packets with ECT are sent, it is important to verify that IP packets with ECN field values of ECT or ECN-CE will reach their destination(s). There is some risk that the use 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 receivers 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 at any time. This is to ensure that packet loss due to ECN marking will not effect the RTCP traffic and the necessary feedback information it carries.

An RTP system that supports ECN MUST implement the initiation of ECN using in-band RTP and RTCP described in Section 7.2.1. It MAY also implement other mechanisms to initiate ECN support, for example the STUN-based mechanism described in Section 7.2.2 or use the leap of faith option if the session supports the limitations provided in Section 7.2.3. If support for both in-band and out-of-band 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 its receivers, to support ECN, following the mechanisms described in Section 7.4. This is necessary because path changes or changes in the receiver population may invalidate the ability of the system to use ECN.

7.2.1. 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 small 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 and RTCP ECN summary reports 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 main 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. A secondary reason to send some not-ECT packets are to ensure that the receivers will send RTCP reports on this sender, even if all ECT marked packets are lost in transit. The not-ECT packets also provide a base-line to compare performance parameters against. A fourth reason for only probing with a small number of packets is to reduce the risk that significant numbers of congestion markings might be lost if ECT is

cleared to Not-ECT by an ECN-Reverting Meddlebox. Then any resulting lack of congestion response is likely to have little damaging affect on others. An RTP sender is RECOMMENDED to send a minimum of two packets with ECT markings per RTCP reporting interval. In case an random ECT pattern is intended to be used, at least one with ECT(0) and one with ECT(1) per reporting interval, in case a single ECT marking is to be used, only that ECT value SHOULD be sent. The RTP sender will continue to send some ECT marked traffic as long as the ECN initiation phase continues. The sender SHOULD 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 sent in duplicate with and without ECT, 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 receivers in the session MUST listen for ECT or ECN-CE marked RTP packets, and generate RTCP ECN feedback packets (Section 5.1) to mark their receipt. An immediate or early (depending on the RTP/AVPF mode) ECN 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. Each regular RTCP report MUST also contain an ECN summary report (Section 5.2). Reception of subsequent ECN-CE marked packets MUST result in additional early or immediate ECN feedback packets being sent unless no timely feedback is required.

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 7.3).

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 is stable, and 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 7.3, 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 any other participants 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., an 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 or ECN summary 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.

If the ECN negotiation succeeds, this indicates that the path can pass some ECN-marked traffic, and that the receivers support ECN

feedback. This does not necessarily imply that the path can robustly convey ECN feedback; Section 7.3 describes the ongoing monitoring that must be performed to ensure the path continues to robustly support ECN.

When a sender or receiver detects ECN failures on paths they should log these to enable follow up and statistics gathering regarding broken paths. The logging mechanism used is implementation dependent.

7.2.2. Detection of ECT using STUN with ICE

This section describes an OPTIONAL method that can be used to avoid media impact and also ensure an ECN capable path prior to media transmission. This method is considered in the context where the session participants are using ICE [RFC5245] 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, an additional connectivity check is performed, sending the "ECT Check" attribute in a STUN packet that is ECT marked. On reception of the packet, a STUN server supporting this extension will note the received ECN field value, and send a STUN/UDP/IP packet in reply, with the ECN field set to not-ECT, and including an ECN check attribute. A STUN server that doesn't understand the extension or is incapable of reading the ECN values on incoming STUN packets should follow the STUN specifications rule for unknown comprehension-optional attributes, i.e. ignore the attribute. Which will result in the sender receiving a STUN response but without the ECN Check STUN attribute.

The STUN ECN check attribute contains one field and a flag. The flag indicates whether 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 attribute is a comprehension optional attribute.

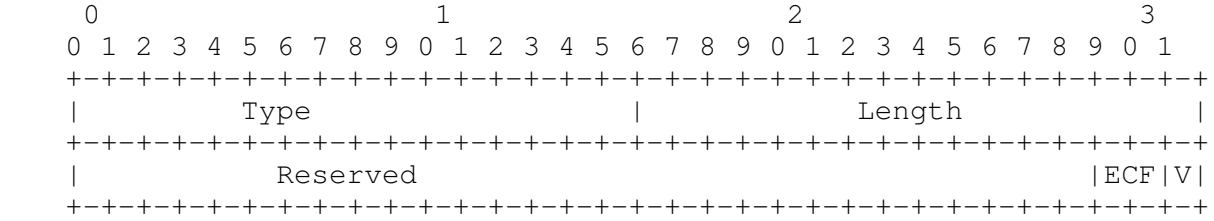


Figure 3: ECN Check STUN Attribute

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 field 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 on transmission, 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 compliant STUN server receiving a request with the ECN Check attribute 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. If the STUN responder was unable to ascertain, due to temporary errors, the ECN value of the STUN request, it SHALL set the V bit in the response to 0. The STUN client may retry immediately.

7.2.3. Leap of Faith ECT initiation method

This method for initiating ECN usage is a leap of faith that assumes that ECN will work on the used path(s). The method is to go directly to "ongoing use of ECN" as defined in Section 7.3. Thus all RTP packets MAY be marked as ECT and the failure detection MUST be used to detect any case when the assumption that the path was ECT capable is wrong. This method is only recommended for controlled environments where the whole path(s) between sender and receiver(s) has been built and verified to be ECT.

If the sender marks all packets as ECT while transmitting on a path that contains an ECN-blocking middlebox, then receivers downstream of that middlebox will not receive any RTP data packets from the sender, and hence will not consider it to be an active RTP SSRC. The sender can detect this and revert to sending packets without ECT marks, since RTCP SR/RR packets from such receivers will either not include

a report for sender's SSRC, or will report that no packets have been received, but this takes at least one RTCP reporting interval. It should be noted that a receiver might generate its first RTCP packet immediately on joining a unicast session, or very shortly after joining a RTP/AVPF session, before it has had chance to receive any data packets. A sender that receives RTCP SR/RR packet indicating lack of reception by a receiver SHOULD therefore wait for a second RTCP report from that receiver to be sure that the lack of reception is due to ECT-marking. Since this recovery process can take several tens of seconds, during which time the RTP session is unusable for media, it is NOT RECOMMENDED that the leap-of-faith ECT initiation method be used in environments where ECN-blocking middleboxes are likely to be present.

7.3. Ongoing Use of ECN Within an RTP Session

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

7.3.1. 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 sender SHOULD mark packets as ECT(0) unless the receiver expresses a preference for ECT(1) or random using the "ect" parameter in the "a=ecn-capable-rtp" attribute.

The sender SHALL NOT include ECT marks on outgoing RTCP packets, and SHOULD NOT include ECT marks on any other 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. For control packets there might be exceptions, like the STUN based ECN check defined in Section 7.2.2.

7.3.2. 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 (subject to the constraints of [RFC4585] and [RFC3550]) to report this back to the sender unless no timely feedback required. There should be no difference in behavior if ECN-CE marks or packet drops are detected. The feedback RTCP packet sent SHALL consist of at least one ECN feedback packet

(Section 5) reporting on the packets received since the last ECN feedback packet, 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, an ECN-capable RTP receiver MUST include an RTCP XR ECN summary report as described in Section 5.2 as part of the compound packet.

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 in a reasonably timely manner, allowing the sender to react on a single or a few reports.

> A possible future optimisation might be to define some form of feedback suppression mechanism to reduce the RTCP reporting overhead for group communication using ECN.

7.3.3. 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 a 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 XR ECN Summary 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 summary information is sufficient (without using RTP/AVPF immediate or early feedback).

Hybrid: There might be mechanisms that utilize both some receiver behaviors and some sender side monitoring, thus requiring both feedback of congestion events to the sender and taking receiver decisions and possible signalling to the sender. From this solution the congestion control algorithm needs to use the signalling to indicate which functions of ECN that is needed to be used.

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 at 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.

We note that ECN support is not a silver bullet to improving performance. The use of ECN gives the chance 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.

7.4. Detecting Failures

Senders and receivers can deliberately ignore ECN-CE and thus get a benefit over behaving flows (cheating). Nonce [RFC3540] is an addition to TCP that solves this issue as long as the sender 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 use of RTP. Still a significant portion of RTP senders are infrastructure devices (for example, streaming media servers) that do have an interest in protecting both service quality and the network. Even though there may be cases where nonce can be applicable also for RTP, it is not included in this specification. This as a receiver interested in cheating would simple claim to not support Nonce. It is however worth mention that, as real-time media is commonly 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 they will affect media quality.

RTP sessions can also suffer from path changes resulting in a non-ECN compliant node becoming 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 sender has sent 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 information or RTCP RR will indicate the situation. The other action is to re-mark a packet from ECT or CE 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.

The ECN feedback packet allows the sender to compare the number of ECT marked packets of different type with the number it actually sent. The number of ECT packets received plus the number of CE marked and lost packets should correspond to the number of sent ECT marked packets unless their is duplication in the network. If this number doesn't agree there are two likely reasons, a translator changing the stream or not carrying the ECN markings forward, or that some node re-marks the packets. In both cases the usage of ECN is broken on the path. By tracking all the different possible ECN field values a sender can quickly detect if some non-compliant behavior is happing on the path.

Thus packet losses and non-matching ECN field value statistics are possible indication of issues with using ECN over the path. The next section defines both sender and receiver reactions to these cases.

7.4.1. 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 ECT-marked packets SHOULD immediately switch to sending packets as not-ECT to determine if the losses potentially are due to the ECT markings. If the losses disappear when the ECT-marking is discontinued, 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 is ECN capable.

2. Renegotiating the session to disable ECN support. This is 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 switch underlying network path in a transparent way for the sender or receiver; and membership changes in a multicast group. It is however appropriate to mention that there are also issues such as re-routing of traffic due to a flappy route table or excessive reordering and other issues that are not directly ECN related but nevertheless may cause problems for ECN.

7.4.2. Interpretation of ECN Summary information

This section contains discussion on how you can use the ECN summary report information in detecting various types of ECN path issues. Lets start to review the information the reports provide on a per source (SSRC) basis:

CE Counter: The number of RTP packets received so far in the session with an ECN field set to CE (11b).

ECT (0/1) Counters: The number of RTP packets received so far in the session with an ECN field set to ECT (0) and ECT (1) respectively (10b / 01b).

not-ECT Counter: The number of RTP packets received so far in the session with an ECN field set to not-ECT (00b)

Lost Packets counter: The number of RTP packets that are expected minus the number received.

Extended Highest Sequence number: The highest sequence number seen when sending this report, but with additional bits, to handle disambiguation when wrapping the RTP sequence number field.

The counters will be initiated to zero to provide value for the RTP stream sender from the very first report. After the first report the changes between the latest received and the previous one is determined by simply taking the values of the latest minus the previous one, taking field wrapping into account. This definition is also robust to packet losses, since if one report is missing, the reporting interval becomes longer, but is otherwise equally valid.

In a perfect world the number of not-ECT packets received should be equal to the number sent minus the lost packets counter, and the sum of the ECT(0), ECT(1), and CE counters should be equal to the number of ECT marked packet sent. Two issues may cause a mismatch in these statistics: severe network congestion or unresponsive congestion control might cause some ECT-marked packets to be lost, and packet duplication might result in some packets being received, and counted in the statistics, multiple times (potentially with a different ECN-mark on each copy of the duplicate).

The level of packet duplication included in the report can be estimated from the sum over all of fields counting received packets compared to the number of packets sent. A high level of packet duplication increases the uncertainty in the statistics, making it more difficult to draw firm conclusions about the behaviour of the network. This issue is also present with standard RTCP reception

reports.

Detecting clearing of ECN field: If the ratio between ECT and not-ECT transmitted in the reports has become all not-ECT or substantially changed towards not-ECT then this is clearly indication that the path results in clearing of the ECT field.

Dropping of ECT packets: To determine if the packet drop ratio is different between not-ECT and ECT marked transmission requires a mix of transmitted traffic. The sender should compare if the delivery percentage (delivered / transmitted) between ECT and not-ECT is significantly different. Care must be taken if the number of packets are low in either of the categories. One must also take into account the level of CE marking. A CE marked packet would have been dropped unless it was ECT marked. Thus, the packet loss level for not-ECT should be aprroximately equal to the loss rate for ECT when counting the CE marked packets as lost ones. A sender performing this calculation needs to ensure that the difference is statistcally significant.

If erronous behavior is detected, it should be logged to enable follow up and statistics gathering.

8. 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. This includes both downstream RTCP reports generated by the media sender, and also reports generated by the receivers, flowing upstream back towards the sender.

8.1. Fragmentation and Reassembly in Translators

An RTP translator may fragment or reassemble RTP data packets without changing the media encoding, and without reference to the congestion state of the networks it bridges. 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 also induce corresponding gaps in the outgoing RTP sequence numbers. An RTP translator MUST rewrite RTCP packets to make the corresponding changes to their sequence numbers, and to reflect the impact of the fragmentation or reassembly. 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.

RTCP ECN feedback packets (Section 5.1) contain six fields that are rewritten in an RTP translator that fragments or reassembles packets: the extended highest sequence number, the lost packets counter, the CE counter, and not-ECT counter, the ECT(0) counter, and the ECT(1) counter. The RTCP XR report block for ECN summary information (Section 5.2) includes a subset of these fields excluding the extended highest sequence number and lost packets counter. The procedures for rewriting these fields are the same for both types of RTCP ECN feedback packet.

When receiving an RTCP ECN feedback packet for the translated stream, an RTP translator first determines the range of packets to which the report corresponds. The extended highest sequence number in the RTCP ECN feedback packet (or in the RTCP SR/RR packet contained within the compound packet, in the case of RTCP XR ECN summary reports) specifies the end sequence number of the range. For the first RTCP ECN feedback packet received, the initial extended sequence number of the range may be determined by subtracting the sum of the lost packets counter, the CE counter, the not-ECT counter, the ECT(0) counter and the ECT(1) counter from the extended highest sequence number (this will be inaccurate if there is packet duplication). For subsequent RTCP ECN feedback packets, the starting sequence number may be determined as being one after the extended highest sequence number of the previous RTCP ECN feedback packet received from the same SSRC. These values are in the sequence number space of the translated packets.

Based on its knowledge of the translation process, the translator determines the sequence number range for the corresponding original, pre-translation, packets. The extended highest sequence number in the RTCP ECN feedback packet is rewritten to match the final sequence number in the pre-translation sequence number range.

The translator then determines the ratio, R, of the number of packets in the translated sequence number space (numTrans) to the number of packets in the pre-translation sequence number space (numOrig) such that R = numTrans / numOrig. The counter values in the RTCP ECN feedback report are then scaled by dividing each of them by R. For example, if the translation process combines two RTP packets into one, then numOrig will be twice numTrans, giving R=0.5, and the counters in the translated RTCP ECN feedback packet will be twice those in the original.

The ratio, R, may have a value that leads to non-integer multiples of the counters when translating the RTCP packet. For example, a VoIP

translator that combines two adjacent RTP packets into one if they contain active speech data, but passes comfort noise packets unchanged, would have an R values of between 0.5 and 1.0 depending on the amount of active speech. Since the counter values in the translated RTCP report are integer values, rounding will be necessary in this case.

When rounding counter values in the translated RTCP packet, the translator should try to ensure that they sum to the number of RTP packets in the pre-translation sequence number space (numOrig). The translator should also try to ensure that no non-zero counter is rounded to a zero value, since that will lose information that a particular type of event has occurred. It is recognised that it may be impossible to satisfy both of these constraints; in such cases, it is better to ensure that no non-zero counter is mapped to a zero value, since this preserves congestion adaptation and helps the RTCP-based ECN initiation process.

It should be noted that scaling the RTCP counter values in this way is meaningful only on the assumption that the level of congestion in the network is related to the number of packets being sent. This is likely to be a reasonable assumption in the type of environment where RTP translators that fragment or reassemble packets are deployed, as their entire purpose is to change the number of packets being sent to adapt to known limitations of the network, but is not necessarily valid in general.

The rewritten RTCP ECN feedback report is sent from the other side of the translator to that which it arrived (as part of a compound RTCP packet containing other translated RTCP packets, where appropriate).

8.2. Generating RTCP ECN Feedback in Media Transcoders

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.

An RTP translator acting as a media transcoder in this manner does not have its own SSRC, and hence is not visible to other entities at the RTP layer. RTCP ECN feedback packets and RTCP XR report blocks

for ECN summary information that are received from downstream relate to the translated stream, and so must be processed by the translator as if it were the original media source. These reports drive the congestion control loop and media adaptation between the translator and the downstream receiver. If there are multiple downstream receivers, a logically separate transcoder instance must be used for each receiver, and must process RTCP ECN feedback and summary reports independently to the other transcoder instances. An RTP translator acting as a media transcoder in this manner MUST NOT forward RTCP ECN feedback packets or RTCP XR ECN summary reports from downstream receivers in the upstream direction.

An RTP translator acting as a media transcoder will generate RTCP reports upstream towards the original media sender, based on the reception quality of the original media stream at the translator. The translator will run a separate congestion control loop and media adaptation between itself and the media sender for each of its downstream receivers, and must generate RTCP ECN feedback packets and RTCP XR ECN summary reports for that congestion control loop using the SSRC of that downstream receiver.

8.3. 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. A mixer has its own SSRC, and is visible to other participants in the session at the RTP layer.

An ECN-aware RTP mixer must generate RTCP ECN feedback packets and RTCP XR report blocks for ECN summary information relating to the RTP flows it terminates, in exactly the same way it would if it were an RTP receiver. These reports form part of the congestion control loop between the mixer and the media senders generating the streams it is mixing. A separate control loop runs between each sender and the mixer.

An ECN-aware RTP mixer will negotiate and initiate the use of ECN on the mixed flows it generates, and will accept and process RTCP ECN feedback reports and RTCP XR report blocks for ECN relating to those mixed flows as if it were a standard media sender. A congestion control loop runs between the mixer and its receivers, driven in part by the ECN reports received.

An RTP mixer MUST NOT forward RTCP ECN feedback packets or RTCP XR ECN summary reports reports from downstream receivers in the upstream direction.

9. 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 pre-dates 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.

10. IANA Considerations

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

10.1. 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.

10.2. RTP/AVPF Transport Layer Feedback Message

The IANA is requested to register one new RTP/AVPF Transport Layer Feedback Message in the table of FMT values for RTPFB Payload Types [RFC4585] as defined in Section 5.1:

Name:	RTCP-ECN-FB
Long name:	RTCP ECN Feedback
Value:	TBA1
Reference:	RFC XXXX

10.3. RTCP Feedback SDP Parameter

The IANA is requested to register one new SDP "rtcp-fb" attribute "nack" parameter "ecn" in the SDP ("ack" and "nack" Attribute Values) registry.

```
Value name:     ecn
Long name:      Explicit Congestion Notification
Usable with:    nack
Reference:      RFC XXXX
```

10.4. RTCP XR Report blocks

The IANA is requested to register one new RTCP XR Block Type as defined in Section 5.2:

```
Block Type: TBA2
Name:       ECN Summary Report
Reference:  RFC XXXX
```

10.5. RTCP XR SDP Parameter

The IANA is requested to register one new RTCP XR SDP Parameter "ecn-sum" in the "RTCP XR SDP Parameters" registry.

Parameter name	XR block (block type and name)
ecn-sum	TBA2 ECN Summary Report Block

10.6. STUN attribute

A new STUN [RFC5389] attribute in the Comprehension-optional range under IETF Review (0x0000 - 0x3FFF) is request to be assigned to the STUN attribute defined in Section 7.2.2. The STUN attribute registry can currently be found at: http://www.iana.org/assignments/stun-parameters/stun-parameters.xhtml.

10.7. ICE Option

A new ICE option "rtp+ecn" is registered in the registry that "IANA Registry for Interactive Connectivity Establishment (ICE) Options" [I-D.ietf-mmusic-ice-options-registry] creates.

11. 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 changing the counter fields. The Issue, 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 re-marking. 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.

Receivers misbehaving: A receiver may prevent the usage of ECN in an RTP session by reporting itself as non ECN capable. 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 on a ECN capable path. 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.

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 [RFC5751] would be used. However, with the limited deployment a minimal mitigation strategy is to require use of SIPS (SIP over TLS) [RFC3261] [RFC5630] to at least accomplish hop-by-hop protection.

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

12. Examples of SDP Signalling

This section contain a few different examples of the signalling mechanism defined in this specification in an SDP context. If there is discrepancies between these examples and the specification text, the specification text is what is correct.

12.1. Basic SDP Offer/Answer

This example is a basic offer/answer SDP exchange, assumed done by SIP (not shown). The intention is to establish a basic audio session point to point between two users.

The Offer:

```
v=0
o=jdoe 3502844782 3502844782 IN IP4 10.0.1.4
s=VoIP call
i=SDP offer for VoIP call with ICE and ECN for RTP
b=AS:128
b=RR:2000
b=RS:2500
a=ice-pwd:YH75Fviy6338Vbrhrlp8Yh
a=ice-ufrag:9uB6
t=0 0
m=audio 45664 RTP/AVPF 97 98 99
c=IN IP4 192.0.2.3
a=rtpmap:97 G719/48000/1
a=fmtp:97 maxred=160
a=rtpmap:98 AMR-WB/16000/1
a=fmtp:98 octet-align=1; mode-change-capability=2
a=rtpmap:99 PCMA/8000/1
a=maxptime:160
a=ptime:20
a=ecn-capable-rtp: ice rtp ect=0 mode=setread
a=rtcp-fb:* nack ecn
a=rtcp-fb:* trr-int 1000
a=rtcp-xr:ecn-sum
a=candidate:1 1 UDP 2130706431 10.0.1.4 8998 typ host
a=candidate:2 1 UDP 1694498815 192.0.2.3 45664 typ srflx raddr
   10.0.1.1 rport 8998
```

This SDP offer offers a single media stream with 3 media payload types. It proposes to use ECN with RTP, with the ICE based initilziation as being prefered over the RTP/RTCP one. Leap of faith is not suggested to be used. The offerer is capable of both setting and reading the ECN bits. In addition the RTCP ECN feedback packet is configured and the RTCP XR ECN summary report. ICE is also proposed with two candidates.

The Answer:

```
v=0
o=jdoe 3502844783 3502844783 IN IP4 198.51.100.235
s=VoIP call
i=SDP offer for VoIP call with ICE and ECN for RTP
b=AS:128
b=RR:2000
b=RS:2500
a=ice-pwd:asd88fgpdd777uzjYhagZg
a=ice-ufrag:8hhY
t=0 0
m=audio 53879 RTP/AVPF 97 99
c=IN IP4 198.51.100.235
a=rtpmap:97 G719/48000/1
a=fmtp:97 maxred=160
a=rtpmap:99 PCMA/8000/1
a=maxptime:160
a=ptime:20
a=ecn-capable-rtp: ice ect=0 mode=readonly
a=rtcp-fb:* nack ecn
a=rtcp-fb:* trr-int 1000
a=rtcp-xr:ecn-sum
a=candidate:1 1 UDP 2130706431 198.51.100.235 53879 typ host
```

The answer confirms that only one media stream will be used. One RTP Payload type was removed. ECN capability was confirmed, and the initilization method will be ICE. However, the answerer is only capable of reading the ECN bits, which means that ECN can only be used for RTP flowing from the offerer to the answerer. ECT always set to 0 will be used in both directions. Both the RTCP ECN feedback packet and the RTCP XR ECN summary report will be used.

12.2. Declarative Multicast SDP

The below session describes an any source multicast using session with a single media stream.

```
v=0
o=jdoe 3502844782 3502844782 IN IP4 198.51.100.235
s=Multicast SDP session using ECN for RTP
i=Multicasted audio chat using ECN for RTP
b=AS:128
t=3502892703 3502910700
m=audio 56144 RTP/AVPF 97
c=IN IP4 224.2.1.3/127
a=rtpmap:97 g719/48000/1
a=fmtp:97 maxred=160
a=maxptime:160
a=ptime:20
a=ecn-capable-rtp: rtp mode=readonly; ect=0
a=rtcp-fb:* nack ecn
a=rtcp-fb:* trr-int 1500
a=rtcp-xr:ecn-sum
```

In the above example, as this is declarative we need to require certain functionality. As it is ASM the initliziation method that can work here is the RTP/RTCP based one. So that is indicated. The ECN setting and reading capability to take part of this session is at least read. If one is capable of setting that is good, but not required as one can skip using ECN for anything one send oneself. The ECT value is recommended to be set to 0 always. The ECN usage in this session requires both ECN feedback and the XR ECN summary report, so their usage are also indicated.

13. Open Issues

As this draft is under development some known open issues exist and are collected here. Please consider them and provide input.

1. The negotiation and directionality attribute is going to need some consideration for multi-party sessions when readonly capability might be sufficient to enable ECN for all incoming streams. However, it would beneficial to know if no potential sender support setting ECN.

2. Consider initiation optimizations that allows for multi SSRC sender nodes to still have rapid usage of ECN.

3. Should we report congestion in bytes or packets? RTCP usually does this in terms of packets, but there may be an argument that we want to report bytes for ECN.
   draft-ietf-tsvwg-byte-pkt-congest is extremely unclear on what is the right approach.

4. We have a saturation problem with the packet loss counters. They do need to continue working even if saturation happens due to long sessions where more lost packets than the counters can handle.

14. References

14.1. Normative References

[I-D.ietf-mmusic-ice-options-registry]
Westerlund, M. and C. Perkins, "IANA Registry for Interactive Connectivity Establishment (ICE) Options", draft-ietf-mmusic-ice-options-registry-00 (work in progress), January 2011.

[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.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", RFC 5245, April 2010.

[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.

14.2. Informative References

[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.zimmermann-avt-zrtp]
Zimmermann, P., Johnston, A., and J. Callas, "ZRTP: Media Path Key Agreement for Unicast Secure RTP", draft-zimmermann-avt-zrtp-22 (work in progress), June 2010.

[RFC2974] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", RFC 2974, October 2000.

[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.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[RFC3569] Bhattacharyya, S., "An Overview of Source-Specific Multicast (SSM)", RFC 3569, July 2003.

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

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, August 2006.

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

[RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, February 2008.

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

[RFC5630] Audet, F., "The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", RFC 5630, October 2009.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

[RFC5760] Ott, J., Chesterfield, J., and E. Schooler, "RTP Control Protocol (RTCP) Extensions for Single-Source Multicast Sessions with Unicast Feedback", RFC 5760, February 2010.

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
School of Computing Science
Glasgow G12 8QQ
United Kingdom

Email: csp@csperkins.org

Piers O'Hanlon
University College London
Computer Science Department
Gower Street
London WC1E 6BT
United Kingdom

Email: p.ohanlon@cs.ucl.ac.uk

Ken Carlberg
G11
1600 Clarendon Blvd
Arlington VA
USA

Email: carlberg@g11.org.uk