Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 29, 2010

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

October 26, 2009

Explicit Congestion Notification (ECN) for RTP over UDP
draft-westerlund-avt-ecn-for-rtp-02

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on April 29, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document specifies how explicit congestion notification (ECN) can be used with RTP/UDP flows that use RTCP as feedback mechanism.

Table of Contents

1. Introduction 3
2. Conventions, Definitions and Acronyms 3
3. Discussion, Requirements, and Design Rationale 4
3.1. Requirements 5
3.2. Applicability 6
4. Use of ECN with RTP/UDP/IP 9
4.1. Negotiation of ECN Capability 12
4.2. Initiation of ECN Use in an RTP Session 16
4.3. Ongoing Use of ECN Within an RTP Session 22
4.4. Detecting Failures and Receiver Misbehaviour 25
5. RTCP Extensions for ECN feedback 29
5.1. ECN Feedback packet 29
5.2. RTCP XR Report block for ECN summary information 32
5.3. RTCP XR Report Block for ECN Nonce 33
6. Processing RTCP ECN Feedback in RTP Translators and Mixers . . 36
6.1. Fragmentation and Reassembly in Translators 36
6.2. Generating RTCP ECN Feedback in Translators 36
6.3. Generating RTCP ECN Feedback in Mixers 36
7. Implementation considerations 37
8. IANA Considerations 37
8.1. SDP Attribute Registration 37
8.2. AVPF Transport Feedback Message 37
8.3. RTCP XR Report blocks 38
8.4. STUN attribute 38
8.5. ICE Option 38
9. Security Considerations 38
10. Examples of SDP Signalling 41
11. Open Issues 41
12. References 41
12.1. Normative References 41
12.2. Informative References 42
Authors' Addresses 44

1. Introduction

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

ECN is getting attention as a method to minimise the impact of congestion on real-time multimedia traffic. 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. This congestion signal allows applications to reduce their transmission rate in a controlled manner, rather than responding to uncontrolled packet loss, and so improves the user experience while benefiting the network.

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 that sender that ECN marked packets are being received, so it can respond to the congestion. TCP [RFC3168], SCTP [RFC4960] and DCCP [RFC4340] have been updated to support ECN, but to date there is no specification how UDP-based transports, such as RTP [RFC3550], can use ECN. This is due to the lack of feedback mechanism directly in UDP. Instead the 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. The means by which ECN is used with RTP over UDP is defined in Section 4, along with RTCP extensions for ECN feedback in Section 5. In Section 6 we discuss how RTCP ECN feedback is handled in RTP translators and mixers. Section 7 discusses some implementation considerations, Section 8 lists IANA considerations, and Section 9 discusses the security considerations.

2. Conventions, Definitions and Acronyms

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Abbreviations

ECN: Explicit Congestion Notification

ECT: ECN Capable Transport

ECN-CE: ECN Congestion Experienced

not-ECT: Not ECN Capable Transport

3. Discussion, Requirements, and Design Rationale

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

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

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

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

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

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

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

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

3.1. Requirements

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

- REQ 1: A mechanism to negotiate and initiate the usage of ECN for RTP/UDP/IP sessions is required

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

- REQ 3: Provide mechanism to minimise the possibility for cheating is desirable

- REQ 4: Some detection and fallback mechanism is needed 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 at the sender by changing the media encoding, at the receiver by changing the subscription to a layered encoding, or in a transcoding middlebox. RFC 5117 identifies seven topologies in which RTP sessions may be configured, and which may affect the ability to use ECN:

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

Topo-Multicast: This is either an any source multicast (ASM) group with potentially several active senders and multicast RTCP feedback, or a source specific multicast (SSM) group with a single sender and unicast RTCP feedback from receivers. RTCP is designed to scale to large group sizes while avoiding feedback implosion (see Section 6.2 of [RFC3550], [RFC4585], and [I-D.ietf-avt-rtcpssm]), and can be used by a sender to determine if all its receivers, and the network paths to those receivers, support ECN (see Section 4.2). It is somewhat more difficult to determine if all network paths from all senders to all receivers support ECN. Accordingly, we allow ECN to be used by an RTP sender using multicast UDP provided the sender has verified that the paths to all its known receivers support ECN, and irrespective of whether the paths from other senders to their receivers support

ECN. Note that group membership may change during the lifetime of a multicast RTP session, potentially introducing new receivers that are not ECN capable. Senders must use the mechanisms described in Section 4.4 to monitor that all receivers continue to support ECN, and needs to fallback to non-ECN use if they do not.

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

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

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

* If the translator is a media transcoder, the output RTP media stream may have radically different characteristics than the input RTP media stream. Each side of the translator must then be considered as a separate transport connection, with its own ECN processing. This requires the translator interpose itself into the ECN negotiation process, effectively splitting the

connection into two parts with their own negotiation. Once negotiation has been completed, the translator must generate RTCP ECN feedback back to the source based on its own reception, and must respond to RTCP ECN feedback received from the receiver(s) (see Section 6.2).

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

Topo-Mixer: 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, 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.

The control of which RTP data packets are marked as ECT, and whether ECT(0) or ECT(1) is used, is due to the sender. RTCP packets must not be ECT marked, whether generated by sender or receivers.

4. 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. Failure detection, verification and fallback

Before an RTP session can be created, a signalling protocol is used to discover the other participants and negotiate session parameters (see Section 4.1). One of the parameters that can be negotiated is the capability of a participant to support ECN functionality, or otherwise. Note that all participants having the capability of supporting ECN does not necessarily imply that ECN is usable in an RTP session, since there may be middleboxes on the path between the participants which don’t 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 4.2):

- 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 also needs to verify ECN support. If verification fails the sender needs to stop using ECN. As the sender will not know of the receiver prior to it sending RTP or RTCP packets, the sender will wait for the first RTCP packet from the new receiver to determine if that contains ECN feedback or not.

- Alternatively, ECN support can be verified during an initial end-to-end STUN exchange (for example, as part of ICE connection establishment). After having verified connectivity without ECN capability an extra STUN exchange, 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 8.4), along with an ICE signalling option (Section 8.5).

- 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 are 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 remarks 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. An RTCP feedback report is sent as soon as possible by the transmission rules for feedback that are in place. This feedback report indicates new CE marks since last ECN feedback packet and also the number of new CE marks through a accumulative 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 Section 4.3 describes the ongoing use of ECN with an RTP session.

This rapid feedback is not optimised for reliability, therefore an additional procedure is used to ensure more reliable, but less timely, reporting of the ECN information. An ECN summary report should also be sent in regular RTCP reports. The ECN summary report contains the same information as the ECN feedback format, only packed differently for better efficiency with large reports. By using accumulative counters for seen CE, ECT, not-ECT or packet loss the sender can determine what events has 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 remarking 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 with those that were sent, while also reporting CE marked and lost packets. If these numbers do not agree with what was sent, it can be inferred that the path does not reliably pass ECN-marked packets

(Section 4.4.2 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.

This specification offers an option of computing and reporting an ECN nonce over all received packets that where not ECN-CE marked or reported explicitly lost. This provides an additional means to detect any packet remarking that happens in the network, and can also be used by a sender to detect receivers that lie about reception of CE-marked packets (it is to be noted that the incentive for receivers to lie in their ECN reports is low for RTP/UDP/IP sessions, since increased congestion levels are likely to cause unpredictable packet losses that decrease the media quality more than would reducing the data rate). To enable the sender to verify the ECN nonce, the sender must learn the sequence number of all packets that was either CE marked or lost, otherwise it can't correctly exclude these packet from the ECN nonce sum. This is done using a new RTCP XR report type, the Nonce Report, that contains the nonce sums and indicating the lost or ECN-CE marked packets using a run length encoded bit-vector. Due to the size of ECN Nonce Reports, and as most RTP-based applications have little incentive to lie about ECN marks, the use of the ECN none is OPTIONAL.

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

4.1. Negotiation of ECN Capability

The first stage of ECN negotiation for RTP-over-UDP is to signal the capability to use ECN. This includes negotiating if ECN is to be used symmetrically, the method for initial ECT verification, and whether the ECN nonce is to be used. 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. In addition an ICE parameter is defined 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 4.1.1. It MAY also implement alternative ECN capability negotiation schemes, such as the ICE extension described in Section 4.1.2.

4.1.1. Signalling ECN Capability using SDP

One new SDP attribute, "a=ecn-capable-rtp", is defined. This is a media level attribute, which MUST NOT be used at the session level. It is not subject to the character set chosen. The aim of this signalling is to indicate the capability of the sender and receivers to support ECN, and to negotiate the method for ECN initiation to be used in the session. Thus the attribute take a list of methods for initiation, which are ordered in decreasing preference. The defined values for the initiation method are:

rtp: Using RTP and RTCP as defined in Section 4.2.1.

ice: Using STUN within ICE as defined in Section 4.2.2.

leap: Using the leap of faith method as defined in Section 4.2.3.

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

- The "mode" 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 downstream congestion has occurred, 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".

- The "nonce" parameter may be used to signal whether the ECN nonce is to be used in the session. This parameter takes two values; "nonce=1" for nonce proposed or shall be used, and "nonce=0" for no nonce.

- The "ect" parameter makes it possible to express the preferred ECT marking. This is either random (default), 0, or 1. If the ECN nonce is used then this parameter MUST be ignored, and random ECT is implied. 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.

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

```
ecn-attribute  = "a=ecn-capable-rtp:" init-list *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     = nonce / mode / ect
mode           = "mode=" ("setonly" / "setread" / "readonly")
nonce          = "nonce=" ("0" / "1")
ect            = "ect=" ("random" / "0" / "1")

; external references:
  ; token: from RFC 4566
```

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 some 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. 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 the "nonce=1" parameter is present in the "a=ecn-capable-rtp" attribute of the offer, the answer MUST explicitly include the "nonce=" parameter in the "a=ecn-capable-rtp" attribute of the answer to indicate if it supports the ECN nonce. If the answer indicates support ("nonce=1") then ECN nonce SHALL be used in the session; if the answer does not include the "nonce=" parameter, or includes "nonce=0", then the ECN nonce SHALL NOT be used. The answer MAY include a "nonce=0" parameter in an answer even if not included in the offer. This indicates that the answerer supports and is interested in using ECN-nonce in this session, but it is not currently enabled. If the offerer supports use of the nonce then it SHOULD run a second round of offer/answer to enable use of the ECN nonce.

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 behaviour of the answering party, and its value in the answer indicates a preference for the behaviour of the offering party. It will be the senders choice if to honor the receivers preference or not.

When SDP is used in a declarative manner, for example in a multicast session using SAP, negotiation of session description parameters is not possible. The "a=ecn-capable-rtp" attribute MAY be added to the session description to indicate that the sender will use ECN in the RTP session. The attribute MUST include a single method of initiation. Participants MUST NOT join such a session unless they

have the capability to understand 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). If the nonce parameter is included the ECN nonce shall be used in the session. The mode parameter MAY be included also in declarative usage, to indicate which 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.

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

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

When using ECN nonce, the RTCP XR signalling indicating the ECN Nonce report MUST also be included in the SDP [RFC3611].

4.1.2. ICE Parameter to Signal ECN Capability

One new ICE [I-D.ietf-mmusic-ice] option, "rtp+ecn", is defined. This is used with the SDP session level "a=ice-options" attribute in an SDP offer to indicate that the initiator of the ICE exchange has the capability to support ECN for RTP-over-UDP flows (via "a=ice-options: rtp+ecn"). The answering party includes this same attribute at the session level in the SDP answer if it 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 4.2.2) 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.

4.2. 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 4.2.1. It MAY also implement other mechanisms to initiate ECN support, for example the STUN-based mechanism described in Section 4.2.2 or use the leap of faith option if the session supports the limitations provided in Section 4.2.3. 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 4.4. This is necessary because path changes or changes in the receiver population may invalidate the ability of the network to support ECN.

4.2.1. Detection of ECT using RTP and RTCP

The ECN initiation phase using RTP and RTCP to detect if the network path supports ECN comprises three stages. Firstly, the RTP sender generates some 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. An RTP sender is RECOMMENDED to send a minimum of two packets with ECT markings per RTCP reporting interval, one with ECT(0) and one with ECT(1), and will continue to send some ECT marked traffic as long as the ECN initiation phase continues. The sender 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 duplicated in transmission, since their RTP sequence number is used to identify packets that are received with ECN markings.

Generating RTCP ECN Feedback: If ECN capability has been negotiated in an RTP session, the 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 SHOULD result in additional early or immediate ECN feedback packets being sent.

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

An RTP sender shall consider the group membership to be stable after it has been in the session and sending ECT-marked probe packets for at least three RTCP reporting intervals (i.e. after sending its third regularly scheduled RTCP packet), and when a complete RTCP reporting interval has passed without changes to the

group membership. ECN initiation is considered successful when the group membership 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 4.3, provided it continues to monitor the RTCP reports for a period of three RTCP reporting intervals from the time the ECN initiation started, to check if there is 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. 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 Section 4.3 describes the ongoing monitoring that must be performed to ensure the path continues to robustly support ECN.

4.2.2. Detection of ECT using STUN with ICE

This section describes an OPTIONAL method that can be used to avoid media impact and also ensure an ECN capable path prior to media transmission. This method is considered in the context where the session participants are using ICE [I-D.ietf-mmusic-ice] to find working connectivity. We need to use ICE rather than STUN only, as the verification needs to happen from the media sender to the address and port on which the receiver is listening.

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

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

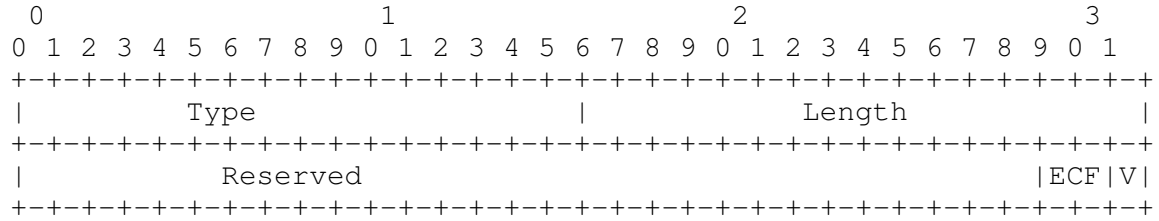


Figure 1: ECN Check Stun Attribute

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

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

Reserved: Reserved bits (29 bits) SHOULD 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 STUN server receiving a request with the ECN Check attribute which understand it SHALL read the ECN field value of the IP/UDP packet the request was received in. Upon forming the response the server SHALL include the ECN Check attribute setting the V bit to valid and include the read value of the ECN field into the ECF field.

4.2.3. 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). It is not generally recommended as the impact on both the application and the network may be substantial if the path is not ECN capable. Applications may experience high packet loss rates, this is both from dropped ECT marked packets, and as a result of driving the network into higher degrees of congestion by not being responsive to ECN marks. The network may experience higher degrees of congestion due to the unresponsiveness of the sender due to lost ECN-CE marks from non-compliant remarking.

The method is to go directly to "ongoing use of ECN" as defined in Section 4.3. 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.

If the sender marks all packets as ECT while transmitting on a path that contains a middlebox that drops all ECT-marked packets, then a receiver downstream of that middlebox will not receive any RTP data packets from that sender, and hence will not consider it to be an active RTP SSRC. The sender can detect this, since SR/RR packets from such receivers will either not include a report for the sender's SSRC, or will include a report claiming that no packets have been received. The sender should be aware that a receiver may 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 may therefore have to wait for a second RTCP report from that receiver to be sure that the lack of reception is due to ECT-marking.

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. It is NOT RECOMMENDED that the leap-of-faith ECT initiation method is used on unmanaged public Internet paths.

4.2.4. ECN Nonce during initiation

If the ECN Nonce was enabled in the signalling, it SHALL be used during the initiation phase as described in Section 4.3.2.1.

4.3. 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.

4.3.1. Transmission of ECT-marked RTP Packets

After a sender has successfully initiated ECN usage, it SHOULD mark all the RTP data packets it sends as ECT. The choice between ECT(0) and ECT(1) is determined by the sender having considered the receiver preferencies expressed by the "ect" parameter in the "a=ecn-capable-rtp" attribute. If the sender selects a random choice of ECT marking, the sender MUST record the statistics for the different ECN values sent. If ECN nonce is activated the sender must record the value and calculate the ECN-nonce sum for outgoing packets [RFC3540] to allow the use of the ECN-nonce to detect receiver misbehaviour (see Section 4.4). Guidelines on the random choice of ECT values are provided in Section 8 of [RFC3540].

The sender SHALL NOT include ECT marks on outgoing RTCP packets, and SHOULD NOT include ECT marks on any 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.

4.3.2. Reporting ECN Feedback via RTCP

An RTP receiver that receives a packet with an ECN-CE mark, or that detects a packet loss, MUST schedule the transmission of an RTCP ECN feedback packet as soon as possible to report this back to the sender. The feedback RTCP packet sent SHALL consist at least one ECN feedback packet (Section 5) reporting on the packets received since the last 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. ECN Nonce information SHOULD NOT be included in early or immediate reports, only when regular reports are sent.

Every time a regular compound RTCP packet is to be transmitted, the RTP receiver MUST include an RTCP XR ECN summary report Section 5.2 as part of the compound packet. If ECN-nonce is enabled the receiver MUST also include an RTCP XR Nonce report packet Section 5.3. It is important to configure the RTCP bandwidth (e.g. using an SDP "b=" line) such that the bit-rate is sufficient for a usage that includes these regular summary and none reports, and feedback on ECN-CE events.

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.

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

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

4.3.2.1. ECN Nonce Reporting

When ECN Nonce reporting is used, it requires both the ECN nonce sum and the sequence numbers for packets where the ECN marking has been lost to be reported. This information is variable size as it depends on both the total number of packet sent per reporting interval and the CE and Packet loss pattern how many bits are required for reporting.

The RTCP packets may be lost, and to avoid the possibility for cheating by "losing" the Nonce information for where one is cheating the nonce coverage needs to be basically complete. Thus the Nonce reporting SHOULD cover at least the 3 regular reporting intervals. The only exception allowed is if the reporting information becomes to heavy and makes the RTCP report packet become larger than the MTU. In that case a receiver MAY reduced to coverage for the ECN nonce to only the last or two last reporting intervals. A sender should consider the received size report for cases where the coverage is not at least three reporting intervals and determine if this may be done to cheat or not. Failure to have reported on all intervals MAY be punished by reducing the congestion safe rate.

The ECN nonce information in the ECN feedback packet consists of both a start value for the nonce prior to the first packet in the reporting interval and the final 2-bit XOR sum over all the received ECN values, both not-ECT and ECT for the report interval. The report interval is explicitly signalled in the RTCP XR Nonce report packet. The initial value for the Nonce is 00b.

4.3.3. Response to Congestion Notifications

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

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

Receiver-Driven Congestion Control: If 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 ECN feedback to the sender, even if it can adapt without contact with the sender, so that the sender can determine if ECN is supported on the network path. The timeliness of RTCP feedback is less of a concern with receiver driven congestion control, and regular RTCP reporting of ECN feedback is sufficient (without using RTP/AVPF immediate or early feedback).

Responding to congestion indication in the case of multicast traffic is a more complex problem than for unicast traffic. The fundamental problem is diverse paths, i.e. when different receivers don't see the same path, and thus have different bottlenecks, so the receivers may get ECN-CE marked packets due to congestion 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.

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

We note that ECN support is not a silver bullet to improving performance. The use of ECN gives the change to respond to congestion before packets are dropped in the network, improving the user experience by allowing the RTP application to control how the quality is reduced. An application which ignores ECN congestion experienced feedback is not immune to congestion: the network will eventually begin to discard packets if traffic doesn't respond. It is in the best interest of an application to respond to ECN congestion feedback promptly, to avoid packet loss.

4.4. Detecting Failures and Receiver Misbehaviour

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

The assumption about the senders acting on the behalf of the network may be reduced due to the nature of peer-to-peer 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. In addition 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 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 remark a packet from ECT to not-ECT. That has less dire results, however, it should be detected so that ECN usage can be suspended to prevent misusing the network.

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. 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 remarks 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.

4.4.1. Fallback mechanisms

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

A receiver that detects a packet loss burst MAY schedule an early feedback packet to report this to the sender that includes at least

the RTCP RR and the ECN feedback message. Thus speeding up the detection at the sender of the losses and thus triggering sender side mitigation.

A sender that detects high packet loss rates for ECT-marked packets SHOULD immediately switch to sender 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 are 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 switches underlying network path in a transparent way for the sender or receiver; and membership changes in a multicast group.

4.4.2. 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 insecurity in the statistics and firm conclusions becomes more difficult and requires clearer statics.

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.

4.4.3. Using ECN-nonce

This document offers ECN Nonce as a method of strengthening the detection of failures, and to allow senders to verify the receiver behavior. We note that it appears counter-productive for a receiver

to attempt to cheat as it most likely will have negative impact on its media quality. However, certain usages of RTP may result in a situation that is more similar to TCP, i.e. where packet losses are repaired and a higher bit-rate is desirable. Thus RTP sessions that use repair mechanisms as FEC or retransmission may consider the usage of the ECN nonce to prevent cheating.

5. RTCP Extensions for ECN feedback

This documents defines three different RTCP extensions: one AVPF NACK Transport feedback format for urgent ECN information; one RTCP XR ECN summary report block type for regular reporting of the ECN marking information; and one additional RTCP XR report block type for ECN nonce.

5.1. ECN Feedback packet

This AVPF NACK 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 the ECN functionality to verify functionality and keep track of when CE marking does occur.

The RTCP AVPF NACK packet starts with the common header defined by the RTP/AVPF profile [RFC4585] which is reproduced here for the reader's information:

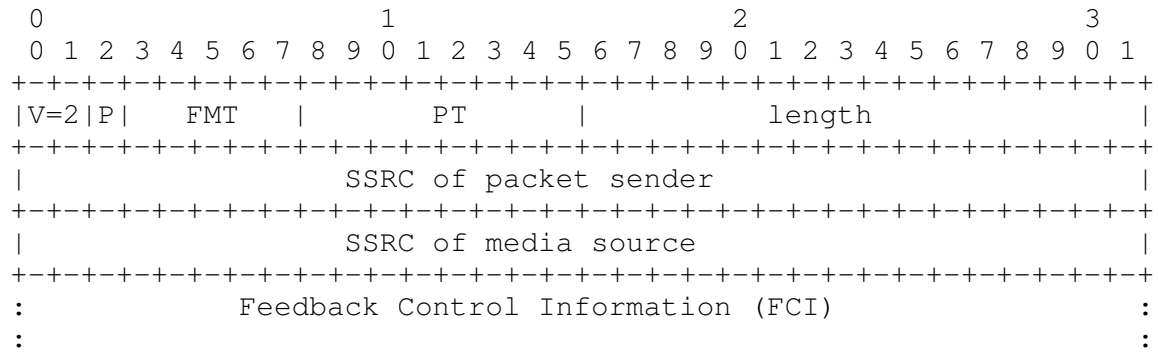


Figure 2: AVPF Feedback common header

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

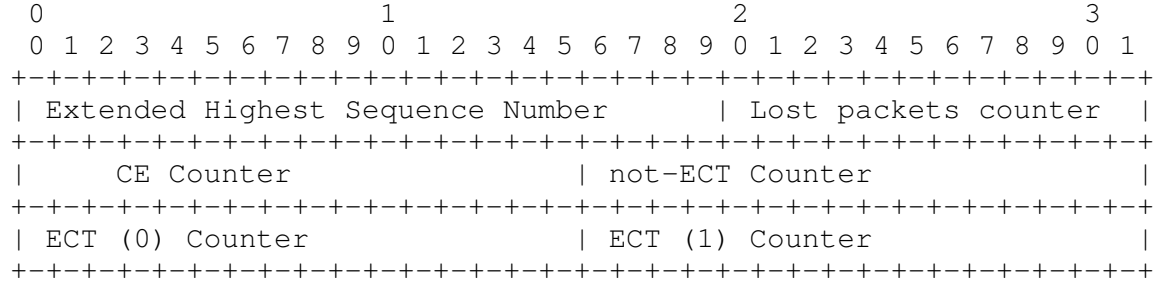


Figure 3: ECN Feedback Format

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

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

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 to 0 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 regular RTCP reporting will include the longer versions of the fields the wrapping issue will be less unless the packet rate of the application is so high that the fields will wrap within a regular RTCP reporting interval. In those case the feedback packet 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.

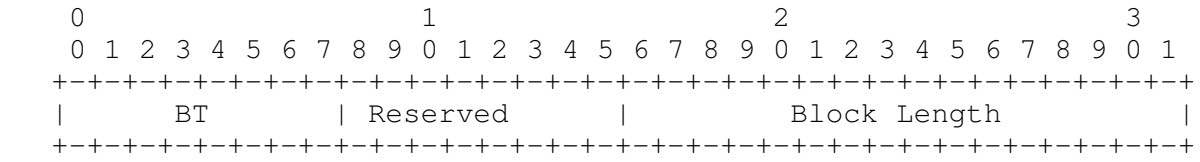
The CE counter is actually more robust for packet duplication. Adding each received CE marked packet to the counter is not an issue.

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.

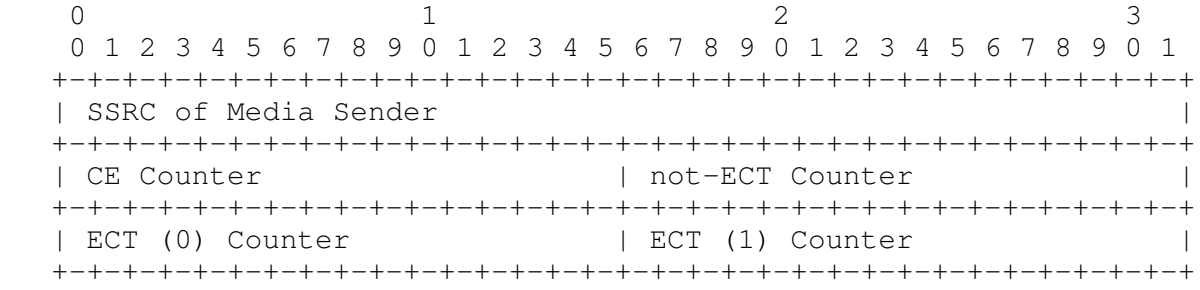
5.2. RTCP XR Report block for ECN summary information

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 always equal 3, 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.

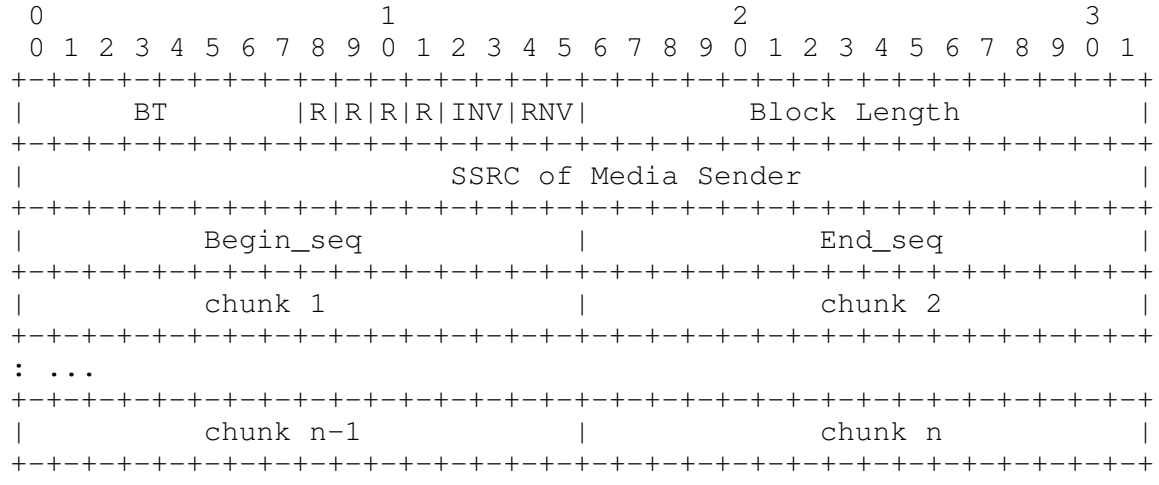
The Extended Highest Sequence number and the packet loss counter for each SSRC is not present in RTCP XR report, in contrast to the feedback version. The reason is that this summary report will always be sent in a RTCP compound packet where the Extended Highest Sequence number and the accumulated number of packet losses are present in the RTCP Sender Report or Receiver Report packet's report block.

5.3. RTCP XR Report Block for ECN Nonce

This RTCP XR block is for ECN Nonce reporting. It consists of an initial part that contains the ECN nonce XOR sum, followed by a series of bit-vector chunks that indicate which RTP sequence numbers were lost or CE-marked, and so weren't included in the ECN nonce sum. The bit-vector uses 1 to indicate that the packet wasn't included in the ECN nonce sum and 0 for packets that where.

The bit-vector is expressed using either Run-Length Encoding or 15-bit explicit bit-vectors. The whole vector is encoded using the 16-bit chunks as defined by Section 4.1.1, 4.1.2, and 4.1.3 in [RFC3611]. The Terminating Null Chunk MUST be used as padding in cases the total number of chunks would otherwise be odd and thus the report block wouldn't reach a 32-bit boundary.

The ECN Nonce report block structure is the following:



BT: Block Type, the value identifying this block is [TBA1].

R: Bits are reserved and MUST be set to 0 on transmission and MUST be ignored on reception.

Block Length: The block length of this full report block in 32-bit words minus one. The minimal report block size is 3, i.e. fixed parts (12 bytes) plus 2 chunks (4 bytes) expressed as 32-bit words (3+1) minus 1.

SSRC of Media Sender SSRC of Media Sender that this report concerns

INV: Initial Nonce Value. Which is the value of Nonce prior to the XOR addition of the ECN field value for the packet that start the nonce reporting interval. This first included sequence number is given by the "begin_seq" value. This to allow running calculations and only need to save nonce values at reporting boundaries.

RNV: Resulting Nonce Value. The Nonce sum value resulting after having XOR the ECN field value for all packets received and not ECN-CE marked with the INV value up to the packet indicated by the "end_seq" sequence number value.

begin_seq: First Sequence number this report covers.

end_seq: Last RTP sequence number included in this report.

chunk i: A chunk reporting on a part of bit-vector indicating if the packet was excluded from the ECN Nonce due to being lost or ECN CE marked.

The Nonce sum initial value for a new media sender (new SSRC) SHALL be 00b. Otherwise the Initial value is the Nonce value calculated for the RTP packet with sequence number begin_seq -1. The initial value for the expressed reporting interval is included in the INV field. The receiver calculate the 2-bit Nonce XOR sum over all received RTP packets in the reporting interval including the one with end_seq sequence number. We note that the RTCP participant doing the Nonce sum MUST perform suppression of packet duplicates. The nonce sum will become incorrect if any duplicates are included in the sum. All packets not received or received as ECN-CE marked when constructing the ECN Nonce report MUST be explicitly marked in the bitvector.

The Nonce reporting interval is RECOMMENDED to cover all the RTP packets received during the three last regular reporting intervals. This is to ensure that the sender will receive a report over all RTP

packets. Failure to deliver reports that cover all the packets may be interpreted as an attempt to cheat.

Two additional considerations must be made when selecting the reporting interval. First, are the MTU considerations. The packet vector and its encoding into chunks results in a variable sized report. The size depends on two main factors, the number of packets to report on and the frequency of bit-value changes in the vector. The reporting interval may need to be shortened to two or even one reporting interval if the resulting ECN nonce report becomes too big to fit into the RTCP packet.

Secondly, the RTP sequence number can easily wrap and that needs to be considered when they are handed. The report SHALL NOT report on more than 32768 consecutive packets. The last sequence number is the extended sequence number that is equal too or smaller (less than 65535 packets) than the value present in the Receiver Reports "extended highest sequence number received" field. The "first sequence number" value is thus an extended sequence number which is smaller than the "last sequence number". If there is a wrap between the first sequence number and the last, i.e. if the first sequence number is greater than the last sequence number (when seen as 16-bit unsigned integers), this needs to included in the calculation. If an application is having these issues, the frequency of regular RTCP reporting should be modified by ensuring that the application chooses appropriate settings for the minimum RTCP reporting interval parameters.

Both the ECN-CE and packet loss information is structured as bit vectors where the first bit represents the RTP packet with the sequence number equal to the First Sequence number. The bit-vector will contain values representing all packets up to and including the one in the "end_seq" field. The chunk mechanism used to represent the bit-vector in an efficient way may appear longer upon reception if an explicit bit-vector is used as the last chunk. Bit-values representing packets with higher sequence number (modulo 16) than "end_seq" are not valid and SHALL be ignored.

The produced bit-vector is encoded using chunks. The chunks are any of the three types defined in [RFC3611], Run Length Chunk (Section 4.1.1 of [RFC3611]), Bit Vector Chunk (Section 4.1.2 of [RFC3611]), or Terminating Null Chunk (Section 4.1.3 of [RFC3611]). Where the Terminating Null Chunk may only appear as the last chunk, and only in cases where the number of chunks otherwise would be odd.

6. Processing RTCP ECN Feedback in RTP Translators and Mixers

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

6.1. Fragmentation and Reassembly in Translators

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

(tbd: complete this section)

6.2. Generating RTCP ECN Feedback in Translators

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

(tbd: complete this section)

6.3. Generating RTCP ECN Feedback in Mixers

An RTP mixer terminates one-or-more RTP flows, combines them into a single outgoing media stream, and transmits that new stream as a separate RTP flow. An ECN-aware RTP mixer must send RTCP reports and provide ECN feedback for the RTP flows it terminates, and must generate RTCP reports for the RTP flow it originates, and add ECT

marks to the outgoing packets. This section describes how RTCP is processed in RTP mixers, and how that interacts with ECN feedback.

(tbd: complete this section)

7. Implementation considerations

To allow the use of ECN with RTP over UDP, the RTP implementation must be able to set the ECT bits in outgoing UDP datagrams, and must be able to read the value of the ECT bits on received UDP datagrams. The standard Berkeley sockets API 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.

8. IANA Considerations

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

8.1. SDP Attribute Registration

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

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

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

8.2. AVPF Transport Feedback Message

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

8.3. RTCP XR Report blocks

Two new RTCP XR report blocks needs to be assigned block type codes.

8.4. STUN attribute

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

8.5. ICE Option

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

9. Security Considerations

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

External threats to the RTP and RTCP traffic:

Denial of Service affecting RTCP: For an attacker that can modify the traffic between the media sender and a receiver can achieve either of two things. 1. Report a lot of packets as being Congestion Experience marked, thus forcing the sender into a congestion response. 2. Ensure that the sender disable the usage of ECN by reporting failures to receive ECN by 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 remarking. It will be mitigated by turning off ECN, however, if the attacker can modify its response to drop packets the same vulnerability exist.

Denial of Service affecting the session set-up signalling: If an attacker can modify the session signalling it can prevent the usage of ECN by removing the signalling attributes used to indicate that the initiator is capable and willing to use ECN with RTP/UDP. This attack can be prevented by authentication and integrity protection of the signalling. We do note that any attacker that can modify the signalling has more interesting attacks they can perform than prevent the usage of ECN, like inserting itself as a middleman in the media flows enabling wire-tapping also for an off-path attacker.

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

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

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

security context to affect the session.

Misbehaving Senders: The enabling of ECN gives the media packets a higher degree of probability to reach the receiver compared to not-ECT marked ones. However, this is no magic bullet and failure to react to congestion will most likely only slightly delay a buffer under-run, in which its session also will experience packet loss and increased delay. There are some chance that the media senders traffic will push other traffic out of the way without being effected to negatively. However, we do note that a media sender still needs to implement congestion control functions to prevent the media from being badly affected by congestion events. Thus the misbehaving sender is getting a unfair share. This can only be detected and potentially prevented by network monitoring and administrative entities. See Section 7 of [RFC3168] for more discussion of this issue.

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

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

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

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

10. Examples of SDP Signalling

(tbd)

11. Open Issues

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

1. Packet duplication. Packet duplication results in uncertainties in the ECN summary counters. At the same time suppressing duplicates and ignoring their ECN marks may also be problematic. Consider the case when a packet get duplicated prior to a congestion point and one version arrives with a ECT mark, and the other with CE mark. What to report?

2. 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 incomming streams. However, it would beneficial to know if no potential sender support setting ECN.

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

4. Feedback suppression for ECN-CE, both for groups, and in case an additional CE mark arrives within a RTT at the receiver.

12. References

12.1. Normative References

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

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

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

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

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

[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.

12.2. Informative References

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

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

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

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

[I-D.ietf-tsvwg-ecn-tunnel] Briscoe, B., "Tunnelling of Explicit Congestion Notification", draft-ietf-tsvwg-ecn-tunnel-03 (work in progress), July 2009.

[I-D.zimmermann-avt-zrtp] Zimmermann, P., Johnston, A., and J. Callas, "ZRTP: Media Path Key Agreement for Secure RTP", draft-zimmermann-avt-zrtp-15 (work in progress), March 2009.

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

June 2002.

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

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

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

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

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

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

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

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

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

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

[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.

Authors' Addresses

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

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

Ingemar Johansson
Ericsson
Laboratoriegrand 11
SE-971 28 Lulea
SWEDEN

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

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

Email: csp@csperkins.org

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