Internet Engineering Task Force (IETF)
Request for Comments: 6679
Category: Standards Track
ISSN: 2070-1721

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

August 2012

Explicit Congestion Notification (ECN) for RTP over UDP

Abstract

This memo specifies how Explicit Congestion Notification (ECN) can be used with the Real-time Transport Protocol (RTP) running over UDP, using the RTP Control Protocol (RTCP) as a feedback mechanism. It defines a new RTCP Extended Report (XR) block for periodic ECN feedback, a new RTCP transport feedback message for timely reporting of congestion events, and a Session Traversal Utilities for NAT (STUN) extension used in the optional initialisation method using Interactive Connectivity Establishment (ICE). Signalling and procedures for negotiation of capabilities and initialisation methods are also defined.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6679.

Table of Contents

1. Introduction

This memo outlines how Explicit Congestion Notification (ECN) [RFC3168] can be used for Real-time Transport Protocol (RTP) [RFC3550] flows running over UDP/IP that use the RTP Control Protocol (RTCP) as a feedback mechanism. The solution consists of feedback of ECN congestion experienced markings to the sender using RTCP, verification of ECN functionality end-to-end, and procedures for how to initiate ECN usage. Since the initiation process has some dependencies on the signalling mechanism used to establish the RTP session, a specification for signalling mechanisms using the Session Description Protocol (SDP) [RFC4566] is included.

ECN can be used to minimise the impact of congestion on real-time multimedia traffic. The use of ECN provides a way for the network to send congestion control signals to the media transport without having to impair the media. Unlike packet loss, ECN signals unambiguously indicate congestion to the transport as quickly as feedback delays allow and without confusing congestion with losses that might have occurred for other reasons such as transmission errors, packet-size errors, routing errors, badly implemented middleboxes, policy violations, and so forth.

The introduction of ECN into the Internet requires changes to both the network and transport layers. At the network layer, IP forwarding has to be updated to allow routers to mark packets, rather than discarding them in times of congestion [RFC3168]. In addition, transport protocols have to be modified to inform the sender that ECN-marked packets are being received, so it can respond to the congestion. The Transmission Control Protocol (TCP) [RFC3168], Stream Control Transmission Protocol (SCTP) [RFC4960], and Datagram Congestion Control Protocol (DCCP) [RFC4340] have been updated to support ECN, but to date, there is no specification describing how UDP-based transports, such as RTP [RFC3550], can use ECN. This is due to the lack of feedback mechanisms in UDP. Instead, the signalling control protocol on top of UDP needs to provide that feedback. For RTP, that feedback is provided by RTCP.

The remainder of this memo is structured as follows. We start by describing the conventions, definitions, and acronyms used in this memo in Section 2 and the design rationale and applicability in Section 3. Section 4 gives an overview of how ECN is used with RTP over UDP. RTCP extensions for ECN feedback are defined in Section 5 and SDP signalling extensions in Section 6. The details of how ECN is used with RTP over UDP are defined in Section 7. In Section 8, we describe how ECN is handled in RTP translators and mixers. Section 9 discusses some implementation considerations; Section 10 lists IANA considerations; and Section 11 discusses security considerations.

Finally, Section 12 provides some examples of SDP signalling for ECN feedback

2. Conventions, Definitions, and Acronyms

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Definitions and Abbreviations:

Sender: A sender of RTP packets carrying an encoded media stream. The sender can change how the media transmission is performed by varying the media coding or packetisation. It is one endpoint of the ECN control loop.

Receiver: A receiver of RTP packets with the intention to consume the media stream. It sends RTCP feedback on the received stream. It is the other endpoint of the ECN control loop.

ECN-Capable Host: A sender or receiver of a media stream that is capable of setting and/or processing ECN marks.

ECN-Capable Transport (ECT): A transport flow where both sender and receiver are ECN-capable hosts. Packets sent by an ECN-capable transport will be marked as ECT(0) or ECT(1) on transmission. See [RFC3168] for the definition of the ECT(0) and ECT(1) marks.

ECN-CE: ECN Congestion Experienced mark (see [RFC3168]).

ECN-Capable Packets: Packets with ECN mark set to either ECT(0), ECT(1), or ECN-CE.

Not-ECT packets: Packets that are not sent by an ECN-capable transport and are not ECN-CE marked.

ECN-Capable Queue: A queue that supports ECN-CE marking of ECN-capable packets to indicate congestion.

ECN-Blocking Middlebox: A middlebox that discards ECN-capable packets.

ECN-Reverting Middlebox: A middlebox that changes ECN-capable packets to not-ECT packets by removing the ECN mark.

Note that RTP mixers or translators that operate in such a manner that they terminate or split the ECN control loop will take on the role of receivers or senders. This is further discussed in Section 3.2.

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 that 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-CE marks are immediately echoed back to the sender by the receiving endpoint 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 setup 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 round-trip time (RTT) (although the RTP Audio-Visual Profile with Feedback (RTP/AVPF) profile [RFC4585] allows more rapid feedback in most cases). RTCP is also very much oriented around counting packets, which makes byte-counting congestion algorithms difficult to utilise.

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 that 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 Any-Source Multicast (ASM) [RFC1112], although a recent extension supports Source-Specific Multicast (SSM) [RFC3569] with unicast feedback [RFC5760]).

Application Awareness: When ECN support is provided within the transport protocol, the ability of the application to react to congestion is limited, since it has little visibility into the transport layer. By adding support of ECN to RTP using RTCP feedback, the application is made aware of congestion, allowing a wider range of reactions in response to that congestion indication.

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

These differences significantly alter the shape of ECN support in RTP over UDP compared to ECN support in TCP, SCTP, and DCCP but do not invalidate the need for ECN support.

ECN support is more important for RTP sessions than, for instance, is the case for many applications over TCP. This is because the impact of packet loss in real-time audio-visual media flows is highly visible to users. For TCP-based applications, however, TCP will retransmit lost packets, and while extra delay is incurred by having packets dropped rather than ECN-CE marked, the loss is repaired. 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 queuing delays in routers, lowering the round-trip time and benefiting interactive applications of RTP, such as voice telephony.

3.1. Requirements

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

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

- REQ 2: A mechanism must exist to feed back the reception of any packets that are ECN-CE marked to the packet sender.

- REQ 3: The provided mechanism should minimise the possibility of cheating (either by the sender or receiver).

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

- REQ 5: Negotiation of ECN should not significantly increase the time taken to negotiate and set up the RTP session (an extra RTT before the media can flow is unlikely to be acceptable for some use cases).

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

The following sections describe how these requirements can be met 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 on the network path(s) traversed. RTP is used in a heterogeneous range of network environments and topologies, with different signalling protocols. The mechanisms defined here make it possible to verify support for ECN in each of these environments, irrespective of the topology.

Due to the need for each RTP sender that intends to use ECN with RTP to track all participants in the RTP session, the sub-sampling of the group membership as specified by "Sampling of the Group Membership in RTP" [RFC2762] MUST NOT be used.

The use of ECN is further dependent on a capability of the RTP media flow to react to congestion signalled by ECN-marked packets. Depending on the application, media codec, and network topology, this adaptation can occur in various forms and at various nodes. As an example, the sender can change the media encoding, the receiver can change the subscription to a layered encoding, or either reaction can be accomplished by a transcoding middlebox. [RFC5117] identifies seven topologies in which RTP sessions may be configured and which may affect the ability to use ECN:

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

Topo-Multicast: This is either an Any-Source Multicast (ASM) group [RFC3569] with potentially several active senders and multicast RTCP feedback or a Source-Specific Multicast (SSM) group [RFC4607] with a single distribution source and unicast RTCP feedback from receivers. RTCP is designed to scale to large group sizes while avoiding feedback implosion (see Section 6.2 of [RFC3550], [RFC4585], and [RFC5760]) and can be used by a sender to determine if all its receivers, and the network paths to those receivers, support ECN (see Section 7.2). It is somewhat more difficult to determine if all network paths from all senders to all receivers support ECN. Accordingly, we allow ECN to be used by an RTP sender using multicast UDP provided the sender has verified that the paths to all its known receivers support ECN, irrespective of whether the paths from other senders to their receivers support ECN ("all its known receivers" are all the synchronisation sources (SSRCs) from which the RTP sender has received RTP or RTCP in the last five reporting intervals, i.e., they have not timed out). Note that group membership may change during the lifetime of a multicast RTP session, potentially introducing new receivers that are not ECN capable or have a path that doesn't support ECN. Senders must use the mechanisms described in Section 7.4 to check that all receivers, and the network paths traversed to reach those receivers, continue to support ECN, and they need to fallback to non-ECN use if any receivers join that do not.

SSM groups that use unicast RTCP feedback [RFC5760] do need a few extra considerations. This topology can have multiple media senders that provide traffic to the distribution source (DS) and are separated from the DS. There can also be multiple feedback targets. The requirement for using ECN for RTP in this topology is that the media sender must be provided the feedback from the receivers. It may be in aggregated form from the feedback targets. We will not mention this SSM use case in the below text

specifically, but when actions are required by the media source, they also apply to the case of SSM where the RTCP feedback goes to the feedback target.

The mechanisms defined in this memo support multicast groups but are known to be conservative and don't scale to large groups. This is primarily because we require all members of the group to demonstrate that they can make use of ECN before the sender is allowed to send ECN-marked packets, since allowing some non-ECN-capable receivers causes fairness issues when the bottleneck link is shared by ECN and non-ECN flows that we have not (yet) been able to satisfactorily address. The rules regarding Determination of ECN Support in Section 7.2.1 may be relaxed in a future version of this specification to improve scaling once these issues have been resolved.

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 translators: those that 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, thus causing ECN negotiation on the path containing the translator to fail (any new RTP translator that does not wish to support ECN may do so similarly). An ECN-aware RTP translator may act in one of three ways:

* If the translator does not modify the media stream, it should copy the ECN bits unchanged from the incoming to the outgoing datagrams, unless it is overloaded and experiencing congestion, in which case it may mark the outgoing datagrams with an ECN-CE mark. Such a translator passes RTCP feedback unchanged. See Section 8.1.

* 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]. See Section 8.2 for details.

* If the translator is a media transcoder, or otherwise modifies the content of the media stream, 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 to interpose itself into the ECN negotiation process, effectively splitting the connection into two parts with their own negotiation. Once negotiation has been completed, the translator must generate RTCP ECN feedback back to the source based on its own reception and must respond to RTCP ECN feedback received from the receiver(s) (see Section 8.3).

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 unicast transport between the mixer and any endpoint 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. See Section 8.4 for details on how mixers should process ECN.

Topo-Video-switch-MCU: A video-switching Multipoint Control Unit (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 SHALL NOT 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 [RFC5117], "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 TCP Friendly Rate Control (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 ECN-CE-marked packets when needed, this mechanism requires support for the RTP/AVPF profile [RFC4585] or any of its derivatives, such as RTP/SAVPF [RFC5124]. The standard RTP/AVP profile [RFC3551] does not allow any early or immediate transmission of RTCP feedback and has a minimal RTCP interval whose default value (5 seconds) is many times the normal RTT between sender and receiver.

3.3. Interoperability

To ensure interoperability for this specification, there is need for at least one common initialisation method for all implementations. Since initialisation using RTP and RTCP (Section 7.2.1) is the one method that works in all cases, although it is not optimal for all uses, it is selected as the mandatory-to-implement initialisation method. This method requires both the RTCP XR extension and the ECN feedback format, which require the RTP/AVPF profile to ensure timely feedback.

When one considers all the uses of ECN for RTP, it is clear that congestion control mechanisms exist that are receiver driven only (Section 7.3.3). These congestion control mechanisms do not require timely feedback of congestion events to the sender. If such a congestion control mechanism is combined with an initialisation method that also doesn’t require timely feedback using RTCP, like the leap-of-faith method (Section 7.2.3) or the ICE-based method (Section 7.2.2), then neither the ECN feedback format nor the RTP/AVPF profile would appear to be needed. However, fault detection can

be greatly improved by using receiver-side detection (Section 7.4.1) and early reporting of such cases using the ECN feedback mechanism.

For interoperability, we mandate the implementation of the RTP/AVPF profile, with both RTCP extensions and the necessary signalling to support a common operations mode. This specification recommends the use of RTP/AVPF in all cases as negotiation of the common interoperability point requires RTP/AVPF, mixed negotiation of RTP/AVP and RTP/AVPF depending on other SDP attributes in the same media block is difficult, and the fact that fault detection can be improved when using RTP/AVPF.

The use of the ECN feedback format is also recommended, but cases exist where its use is not required because timely feedback is not needed. These will be explicitly noted using the phrase "no timely feedback required" and generally occur in combination with receiver-driven congestion control and with the leap-of-faith and ICE-based initialisation methods. We also note that any receiver-driven congestion control solution that still requires RTCP for signalling of any adaptation information to the sender will still require RTP/AVPF for timeliness.

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

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

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

2. Initiation and initial verification of ECN-capable transport

3. Ongoing use of ECN within an RTP session

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

Before an RTP session can be created, a signalling protocol is used to negotiate or at least configure session parameters (see Section 7.1). In some topologies, the signalling protocol can also be used to discover the other participants. One of the parameters that must be agreed is the capability of a participant to support ECN. 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 that 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 to support ECN, it needs to verify that the ECN support is usable. There are three ways in which this verification can be done:

- The sender may generate a (small) subset of its RTP data packets with the ECN field of the IP header 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. When a new receiver joins an existing RTP session, it will send RTCP reports in the usual manner. If those RTCP reports include ECN information, verification will have succeeded, and sources can continue to send ECT packets. If not, verification fails, and each sender MUST stop using ECN (see Section 7.2.1 for details).

- 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 on the candidate path that is about to be used. If successful, the path's capability to convey ECN-marked packets is verified. A new STUN attribute is defined to convey feedback that the ECT-marked STUN request was received (see Section 7.2.2), along with an ICE signalling option (Section 6.4) to indicate that the check is to be performed.

- Thirdly, the sender may make a leap of faith that ECN will work. This is only recommended for applications that know they are running in controlled environments where ECN functionality has been verified through other means. In this mode, it is assumed that ECN works, and the system reacts to failure indicators if the assumption proved wrong. The use of this method relies on a high confidence that ECN operation will be successful or an application where failure is not serious. The impact on the network and other users must be considered when making a leap of faith, so there are limitations on when this method is allowed (see Section 7.2.3).

The first mechanism, using RTP with RTCP feedback, has the advantage of working for all RTP sessions, but the disadvantages of potential clipping if ECN-marked RTP packets are discarded by middleboxes and slow verification of ECN support. The STUN-based mechanism is faster to verify ECN support but only works in those scenarios supported by end-to-end STUN, such as within an ICE exchange. The third one, leap of faith, has the advantage of avoiding additional tests or complexities and enabling ECN usage from the first media packet. The downside is that if the end-to-end path contains middleboxes that do

not pass ECN, the impact on the application can be severe: in the worst case, all media could be lost if a middlebox that discards ECN-marked packets is present. A less severe effect, but still requiring reaction, is the presence of a middlebox that re-marks ECT-marked packets to not-ECT, possibly marking packets with an ECN-CE mark as not-ECT. This could result in increased levels of congestion due to non-responsiveness and impact media quality as applications end up relying on packet loss as an indication of congestion.

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 feed back reports on any ECN-CE marks to the sender using RTCP in RTP/AVPF immediate or early feedback mode, unless no timely feedback is required. Each feedback report indicates the receipt of new ECN-CE marks since the last ECN feedback packet and also counts the total number of ECN-CE-marked packets as a cumulative sum. This is the mechanism to provide the fastest possible feedback to senders about ECN-CE marks. On receipt of an ECN-CE-marked packet, the system must react to congestion as if packet loss has been reported. Section 7.3 describes the ongoing use of ECN within an RTP session.

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

RTCP reports MUST NOT be ECT marked, since ECT-marked traffic may be dropped if the path is not ECN compliant. RTCP is used to provide feedback about what has been transmitted and what ECN markings that are received, so it is important that it is 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 and link failure followed by re-routing around it. Such an event may result in the packet being sent through a node that is ECN non-compliant, thus re-marking or dropping packets with ECT set. To prevent this from impacting the application for longer than necessary, the operation of ECN is constantly monitored by all senders (Section 7.4). Both the RTCP XR ECN Summary Reports and the ECN feedback packets allow the sender to compare the number of ECT(0), ECT(1), and not-ECT-marked

packets received with the number that were sent, while also reporting ECN-CE-marked and lost packets. If these numbers do not agree, it can be inferred that the path does not reliably pass ECN-marked packets. 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.

5. RTCP Extensions for ECN Feedback

This memo defines two new RTCP extensions: one RTP/AVPF [RFC4585] transport-layer feedback format for reporting urgent ECN information and one RTCP XR [RFC3611] ECN Summary Report block type for regular reporting of the ECN marking information.

5.1. RTP/AVPF Transport-Layer ECN Feedback Packet

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

The RTP/AVPF transport-layer feedback packet starts with the common header defined by the RTP/AVPF profile [RFC4585], which is reproduced in Figure 1. The FMT field takes the value 8 to indicate that the Feedback Control Information (FCI) contains an ECN Feedback Report, as defined in Figure 2.

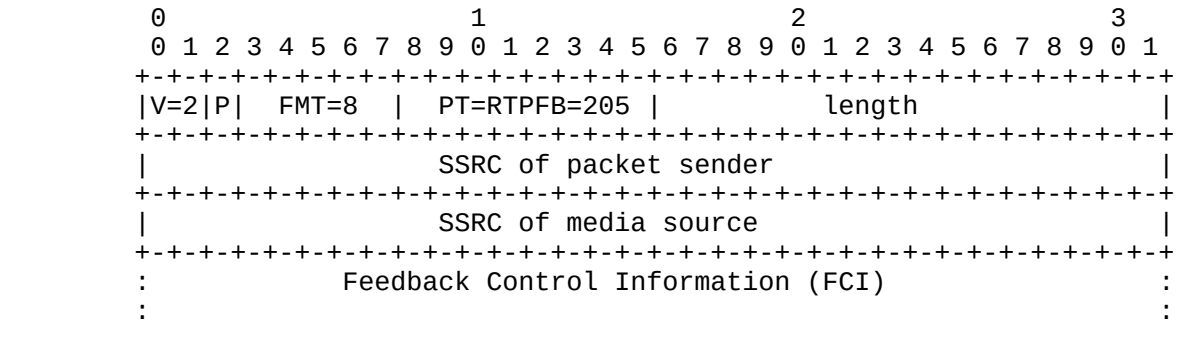

Figure 1: RTP/AVPF Common Packet Format for Feedback Messages

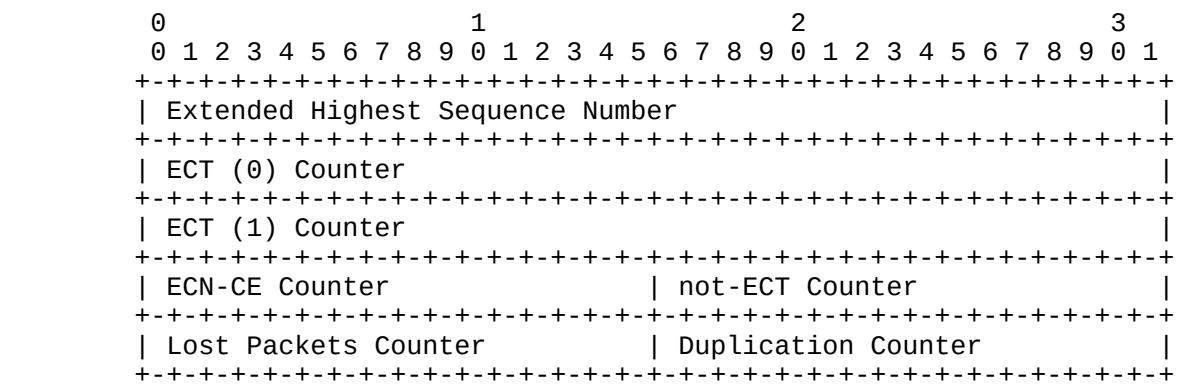

Figure 2: ECN Feedback Report Format

The ECN Feedback Report contains the following fields:

Extended Highest Sequence Number: The 32-bit extended highest sequence number received, as defined by [RFC3550]. Indicates the highest RTP sequence number to which this report relates.

ECT(0) Counter: The 32-bit cumulative number of RTP packets with ECT(0) received from this SSRC.

ECT(1) Counter: The 32-bit cumulative number of RTP packets with ECT(1) received from this SSRC.

ECN-CE Counter: The cumulative number of RTP packets received from this SSRC since the receiver joined the RTP session that were ECN-CE marked, including ECN-CE marks in any duplicate packets. The receiver should keep track of this value using a local representation that is at least 32 bits and only include the 16

bits with least significance. In other words, the field will wrap if more than 65535 ECN-CE-marked 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 at least 32 bits and only include the 16 bits with least significance. In other words, the field will wrap if more than 65535 not-ECT packets have been received.

Lost Packets Counter: The cumulative number of RTP packets that the receiver expected to receive minus the number of packets it actually received that are not a duplicate of an already received packet, from this SSRC since the receiver joined the RTP session. Note that packets that arrive late are not counted as lost. The receiver should keep track of this value using a local representation that is at least 32 bits and only include the 16 bits with least significance. In other words, the field will wrap if more than 65535 packets are lost.

Duplication Counter: The cumulative number of RTP packets received that are a duplicate of an already received packet from this SSRC since the receiver joined the RTP session. The receiver should keep track of this value using a local representation that is at least 32 bits and only include the 16 bits with least significance. In other words, the field will wrap if more than 65535 duplicate packets have been received.

All fields in the ECN Feedback Report are unsigned integers in network byte order. Each ECN Feedback Report corresponds to a single RTP source (SSRC). Multiple sources can be reported by including multiple ECN Feedback Report packets in an compound RTCP packet.

The counters SHALL be initiated to 0 for each new SSRC received. This enables detection of ECN-CE marks or packet loss on the initial report from a specific participant.

The use of at least 32-bit counters allows even extremely high packet volume applications to not have wrapping of counters within any timescale close to the RTCP reporting intervals. However, 32 bits are not sufficiently large to disregard the fact that wrappings may happen during the lifetime of a long-lived RTP session, and implementations need to be written to handle wrapping of the counters. It is recommended that implementations use local representation of these counters that are longer than 32 bits to enable easy handling of wraps.

There is a difference in packet duplication reports between the packet loss counter that is defined in the Receiver Report Block [RFC3550] and that defined here. To avoid holding state for what RTP sequence numbers have been received, [RFC3550] specifies that one can count packet loss by counting the number of received packets and comparing that to the number of packets expected. As a result, a packet duplication can hide a packet loss. However, when populating the ECN Feedback Report, a receiver needs to track the sequence numbers actually received and count duplicates and packet loss separately to provide a more reliable indication. Reordering may, however, still result in packet loss being reported in one report and then removed in the next.

The ECN-CE counter is robust for packet duplication. Adding each received ECN-CE-marked packet to the counter is not an issue; in fact, it is required to ensure complete tracking of the ECN state. If one of the clones was ECN-CE marked, that is still an indication of congestion. Packet duplication has a potential impact on the ECN verification, and there is thus a need to count the duplicates.

5.2. RTCP XR Report Block for ECN Summary Information

This unilateral XR report block combined with RTCP SR or RR report blocks carries the same information as the ECN Feedback Report and is based on the same underlying information. However, the ECN Feedback Report is intended to report an ECN-CE mark as soon as possible, while this extended report is for the regular RTCP reporting and continuous verification of the ECN functionality end-to-end.

The ECN Summary Report block consists of one RTCP XR report block header, shown in Figure 3 followed by one or more ECN Summary Report data blocks, as defined in Figure 4.

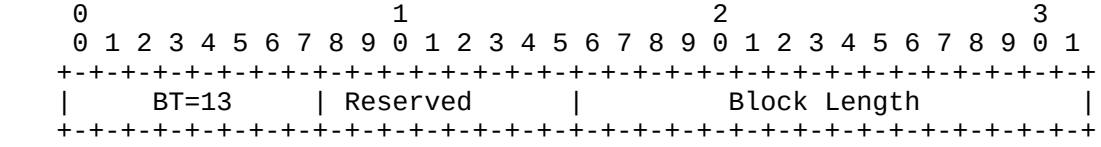

Figure 3: RTCP XR Report Header

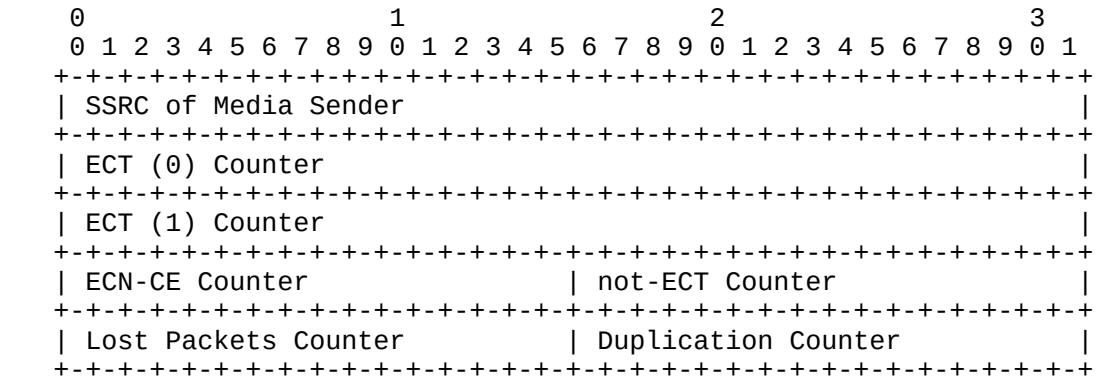

Figure 4: RTCP XR ECN Summary Report

The RTCP XR ECN Summary Report contains the following fields:

BT: Block Type identifying the ECN Summary Report block. Value is 13.

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

Block Length: The length of this XR report block, including the header, in 32-bit words minus one. Used to indicate the number of ECN Summary Report data blocks present in the ECN Summary Report. This length will be 5*n, where n is the number of ECN Summary Report blocks, since blocks are a fixed size. The block length MAY be zero if there is nothing to report. Receivers MUST discard reports where the block length is not a multiple of five, since these cannot be valid.

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

ECT(0) Counter: as in Section 5.1.

ECT(1) Counter: as in Section 5.1.

ECN-CE Counter: as in Section 5.1.

not-ECT Counter: as in Section 5.1.

Lost Packets Counter: as in Section 5.1.

Duplication Counter: as in Section 5.1.

The extended highest sequence number counter for each SSRC is not present in an RTCP XR report, in contrast to the feedback version. The reason is that this summary report will rely on the information sent in the Sender Report (SR) or Receiver Report (RR) blocks part of the same RTCP compound packet. The extended highest sequence number is available from the SR or RR.

All the SSRCs that are present in the SR or RR SHOULD also be included in the RTCP XR ECN Summary Report. In cases where the number of senders are so large that the combination of SR/RR and the ECN summary for all the senders exceed the MTU, then only a subset of the senders SHOULD be included so that the reports for the subset fits within the MTU. The subsets SHOULD be selected round-robin across multiple intervals so that all sources are periodically reported. In case there are no SSRCs that currently are counted as senders in the session, the report block SHALL still be sent with no report block entry and a zero report block length to continuously indicate to the other participants the receiver capability to report ECN information.

6. SDP Signalling Extensions for ECN

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

6.1. Signalling ECN Capability Using SDP

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

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

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

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

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

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

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

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

The ABNF [RFC5234] grammar for the "a=ecn-capable-rtp:" attribute is shown in Figure 5.

```
ecn-attribute  = "a=ecn-capable-rtp:" SP init-list [SP parm-list]
init-list      = init-value *("," init-value)
init-value     = "rtp" / "ice" / "leap" / init-ext
init-ext       = token
parm-list      = parm-value *(";" SP parm-value)
parm-value     = mode / ect / parm-ext
mode           = "mode=" ("setonly" / "setread" / "readonly")
ect            = "ect=" ("0" / "1" / "random")
parm-ext       = parm-name "=" parm-value-ext
parm-name      = token
parm-value-ext = token / quoted-string
quoted-string = ( DQUOTE *qdtext DQUOTE )
qdtext = %x20-21 / %x23-5B / %x5D-7E / quoted-pair / UTF8-NONASCII
   ; No DQUOTE and no "\"
quoted-pair = "\\" / ( "\" DQUOTE )
UTF8-NONASCII = UTF8-1 / UTF8-2 / UTF8-3 / UTF8-4

; external references:
  ; token from RFC 4566
  ; SP and DQUOTE from RFC 5234
  ; UTF8-1, UTF8-2, UTF8-3, and UTF8-4 from RFC 3629
```

Figure 5: ABNF Grammar for the "a=ecn-capable-rtp:" Attribute

Note the above quoted string construct has an escaping mechanism for strings containing ". This uses \ (backslash) as an escaping mechanism, i.e., a " is replaced by \" (backslash double quote) and any \ (backslash) is replaced by \\ (backslash backslash) when put into the double quotes as defined by the above syntax. The string in a quoted string is UTF-8 [RFC3629].

6.1.1. Use of "a=ecn-capable-rtp:" with the Offer/Answer Model

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 session for which it wishes to use ECN. The attribute includes one or more ECN initiation methods in a comma-separated list in decreasing order of preference, with any number of optional parameters following. The answering party compares the list of initiation methods in the offer with those it supports in order of preference. If there is a match and if the receiver wishes to attempt to use ECN in the session, it includes an "a=ecn-capable-rtp:" attribute containing its single preferred choice of initiation method, and any optional parameters, in the media sections of the answer. If there is no matching

initiation method capability, or if the receiver does not wish to attempt to use ECN in the session, it does not include an "a=ecn--capable-rtp:" attribute in its answer. If the attribute is removed in the answer, then ECN MUST NOT be used in any direction for that media flow. If there are initialisation methods that are unknown, they MUST be ignored on reception and MUST NOT be included in an answer.

The endpoints' capability to set and read ECN marks, as expressed by the optional "mode=" parameter, determines whether ECN support can be negotiated for flows in one or both directions:

- 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.

- An offer that does not include a "mode=" parameter MUST be treated as if a "mode=setread" parameter had been included.

In an RTP session using multicast and ECN, participants that intend to send RTP packets SHOULD support setting ECT marks in RTP packets (i.e., should be "mode=setonly" or "mode=setread"). Participants receiving data need the capability to read ECN marks on incoming packets. It is important that receivers can read ECN marks ("mode=readonly" or "mode=setread"), since otherwise no sender in the

multicast session would be able to enable ECN. Accordingly, receivers that are "mode=setonly" SHOULD NOT join multicast RTP sessions that use ECN. If session participants that are not aware of the ECN for RTP signalling are invited to a multicast session and simply ignore the signalling attribute, the other party in the offer/answer exchange SHOULD terminate the SDP dialogue so that the participant leaves the session.

The "ect=" parameter in the "a=ecn-capable-rtp:" attribute is set independently in the offer and the answer. Its value in the offer indicates a preference for the sending behaviour of the answering party, and its value in the answer indicates a sending preference for the behaviour of the offering party. It will be the sender's choice to honour the receiver's preference for what to receive or not. In multicast sessions, all senders SHOULD set the ECT marks using the value declared in the "ect=" parameter.

Unknown optional parameters MUST be ignored on reception and MUST NOT be included in the answer. That way, a new parameter may be introduced and verified as supported by the other endpoint by having the endpoint include it in any answer.

6.1.2. Use of "a=ecn-capable-rtp:" with Declarative SDP

When SDP is used in a declarative manner, for example, in a multicast session using the Session Announcement Protocol (SAP) [RFC2974], negotiation of session description parameters is not possible. The "a=ecn-capable-rtp:" attribute MAY be added to the session description to indicate that the sender will use ECN in the RTP session. The attribute MUST include a single method of initiation. Participants MUST NOT join such a session unless they have the capability to receive ECN-marked UDP packets, implement the method of initiation, and generate RTCP ECN feedback. The mode parameter MAY also be included in declarative usage, to indicate the minimal capability is required by the consumer of the SDP. So, for example, in an SSM session, the participants configured with a particular SDP will all be in a media receive-only mode; thus, "mode=readonly" may be used as the receiver only needs to be able to report on the ECN markings. In ASM sessions, using "mode=readonly" is also reasonable, unless all senders are required to attempt to use ECN for their outgoing RTP data traffic, in which case the mode needs to be set to "setread".

6.1.3. General Use of the "a=ecn-capable-rtp:" Attribute

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

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

6.2. RTCP ECN Feedback SDP Parameter

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

```
rtcp-fb-nack-param  =  <See Section 4.2 of [RFC4585]>
rtcp-fb-nack-param  =/ ecn-fb-par
ecn-fb-par          =  SP "ecn"
```

The offer/answer rules for these SDP feedback parameters are specified in the RTP/AVPF profile [RFC4585].

6.3. XR Block ECN SDP Parameter

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

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

For SDP declarative and offer/answer usage, see the RTCP XR specification [RFC3611] and its description of how to handle unilateral parameters.

6.4. ICE Parameter to Signal ECN Capability

One new ICE [RFC5245] option, "rtp+ecn", is defined. This is used with the SDP session level "a=ice-options" attribute in an SDP offer to indicate that the initiator of the ICE exchange has the capability to support ECN for RTP-over-UDP flows (via "a=ice-options: rtp+ecn"). The answering party includes this same attribute at the session level in the SDP answer if it also has the capability and removes the attribute if it does not wish to use ECN or doesn't have the capability to use ECN. If the ICE initiation method (Section 7.2.2) is actually going to be used, it is also needs to be explicitly negotiated using the "a=ecn-capable-rtp:" attribute. This ICE option SHALL be included when the ICE initiation method is offered or declared in the SDP.

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

7. Use of ECN with RTP/UDP/IP

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

7.1. Negotiation of ECN Capability

The first stage of ECN negotiation for RTP over UDP is to signal the capability to use ECN. An RTP system that supports ECN and uses SDP for its signalling MUST implement the SDP extension to signal ECN capability as described in Section 6.1, the RTCP ECN feedback SDP parameter defined in Section 6.2, and the XR Block ECN SDP parameter defined in Section 6.3. It MAY also implement alternative ECN capability negotiation schemes, such as the ICE extension described in Section 6.4. Other signalling systems will need to define signalling parameters corresponding to those defined for SDP.

The "ecn-capable-rtp:" SDP attribute MUST be used when employing ECN for RTP according to this specification in systems using SDP. As the RTCP XR ECN Summary Report is required independently of the initialisation method or congestion control scheme, the "rtcp-xr" attribute with the "ecn-sum" parameter MUST also be used. The "rtcp-fb" attribute with the "nack" parameter "ecn" MUST be used whenever the initialisation method or a congestion control algorithm

requires timely sender-side knowledge of received CE markings. If the congestion control scheme requires additional signalling, this should be indicated as appropriate.

7.2. Initiation of ECN Use in an RTP Session

Once the sender and the receiver(s) have agreed that they have the capability to use ECN within a session, they may attempt to initiate ECN use. All session participants connected over the same transport MUST use the same initiation method. RTP mixers or translators can use different initiation methods to different participants that are connected over different underlying transports. The mixer or translator will need to do individual signalling with each participant to ensure it is consistent with the ECN support in those cases where it does not function as one endpoint for the ECN control loop.

At the start of the RTP session, when the first few 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, the sender needs to stop using ECN immediately to protect both the network and the application.

The RTP senders and receivers SHALL NOT ECT mark their RTCP traffic at any time. This is to ensure that packet loss due to ECN marking will not effect the RTCP traffic and the necessary feedback information it carries.

An RTP system that supports ECN MUST implement the initiation of ECN using in-band RTP and RTCP described in Section 7.2.1. It MAY also implement other mechanisms to initiate ECN support, for example, the STUN-based mechanism described in Section 7.2.2, or use the leap-of-faith option if the session supports the limitations provided in Section 7.2.3. If support for both in-band and out-of-band mechanisms is signalled, the sender when negotiating SHOULD offer detection of ECT using STUN with ICE with higher priority than detection of ECT using RTP and RTCP.

No matter how ECN usage is initiated, the sender MUST continually monitor the ability of the network, and all its receivers, to support ECN, following the mechanisms described in Section 7.4. This is necessary because path changes or changes in the receiver population may invalidate the ability of the system to use ECN.

7.2.1. Detection of ECT Using RTP and RTCP

The ECN initiation phase using RTP and RTCP to detect if the network path supports ECN comprises three stages. First, the RTP sender generates some small fraction of its traffic with ECT marks to act as 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 is 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 baseline to compare performance parameters against. Another reason for only probing with a small number of packets is to reduce the risk that significant numbers of congestion markings might be lost if ECT is cleared to not-ECT by an ECN-reverting Middlebox. Then, any resulting lack of congestion response is likely to have little damaging effect on others. An RTP sender is RECOMMENDED to send a minimum of two packets with ECT markings per RTCP reporting interval. In case a random ECT pattern is intended to be used, at least one packet with ECT(0) and one with ECT(1) should be sent per reporting interval; in case a single ECT marking is to be used, only that ECT value SHOULD be sent. The RTP sender SHALL 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 very media dependent. For audio formats, it would make sense for the sender to mark comfort noise packets or similar. For video formats, packets containing P- or B-frames (rather than I-frames) would be an appropriate choice. No matter which RTP packets are marked, those packets MUST NOT be sent in duplicate, with and without ECT, since the RTP sequence number is used to identify packets that are received with ECN markings.

Generating RTCP ECN Feedback: If ECN capability has been negotiated in an RTP session, the receivers in the session MUST listen for ECT or ECN-CE-marked RTP packets and generate RTCP ECN feedback packets (Section 5.1) to mark their receipt. An immediate or early (depending on the RTP/AVPF mode) ECN feedback packet SHOULD be generated on receipt of the first ECT- or ECN-CE-marked packet from a sender that has not previously sent any ECT traffic. Each regular RTCP report MUST also contain an ECN Summary Report (Section 5.2). Reception of subsequent ECN-CE-marked packets MUST result in additional early or immediate ECN feedback packets being sent unless no timely feedback is required.

Determination of ECN Support: RTP is a group communication protocol, where members can join and leave the group at any time. This complicates the ECN initiation phase, since the sender must wait until it believes the group membership has stabilised before it can determine if the paths to all receivers support ECN (group membership changes after the ECN initiation phase has completed are discussed in Section 7.3).

An RTP sender shall consider the group membership to be stable after it has been in the session and sending ECT-marked probe packets for at least three RTCP reporting intervals (i.e., after sending its third regularly scheduled RTCP packet) and when a complete RTCP reporting interval has passed without changes to the group membership. ECN initiation is considered successful when the group membership is stable and all known participants have sent one or more RTCP ECN feedback packets or RTCP XR ECN Summary Reports 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 an RTCP ECN Feedback Report or an RTCP XR ECN Summary Report indicating successful receipt of the ECT-marked packets, with no negative indications, from a single RTP receiver (where a single RTP receiver is considered as all SSRCs used by a single RTCP CNAME). After declaring provisional success, the sender MAY generate ECT-marked packets as described in Section 7.3, provided it continues to monitor the RTCP reports for a period of three RTCP reporting intervals from the time the ECN initiation started, to check if there are any other participants in the session. Thus, as long as any additional SSRC that report on the ECN usage are using the same RTCP CNAME as the previous reports and they are all indicating functional ECN, the sender may continue. If other participants are detected, i.e., other RTCP CNAMEs, 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. Different RTCP CNAMEs received over a unicast transport may occur when using translators in a multi-party RTP session (e.g., when using a centralised conference bridge).

Note: The above optimisation supports peer-to-peer unicast transport with several SSRCs multiplexed onto the same flow (e.g., a single participant with two video cameras or SSRC multiplexed RTP retransmission [RFC4588]). It is desirable to be able to rapidly negotiate ECN support for such a session, but the optimisation above can fail if there are implementations that use the same CNAME for different parts of a distributed implementation that have different transport characteristics (e.g., if a single logical endpoint is split across multiple hosts).

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

If the ECN negotiation succeeds, this indicates that the path can pass some ECN-marked traffic and that the receivers support ECN feedback. This does not necessarily imply that the path can robustly convey ECN feedback; Section 7.3 describes the ongoing monitoring that must be performed to ensure the path continues to robustly support ECN.

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

7.2.2. Detection of ECT Using STUN with ICE

This section describes an OPTIONAL method that can be used to avoid media impact and also ensure an ECN-capable path prior to media transmission. This method is considered in the context where the session participants are using ICE [RFC5245] to find working connectivity. We need to use ICE rather than STUN only, as the verification needs to happen from the media sender to the address and port on which the receiver is listening.

Note that this method is only applicable to sessions when the remote destinations are unicast addresses. In addition, transport translators that do not terminate the ECN control loop and may distribute received packets to more than one other receiver must either disallow this method (and use the RTP/RTCP method instead) or implement additional handling as discussed below. This is because the ICE initialisation method verifies the underlying transport to one particular address and port. If the receiver at that address and port intends to use the received packets in a multi-point session, then the tested capabilities and the actual session behaviour are not matched.

To minimise the impact of setup delay, and to prioritise the fact that one has 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, an additional connectivity check is performed, sending the "ECN-CHECK" attribute in a STUN packet that is ECT marked. On reception of the packet, a STUN server supporting this extension will note the received ECN field value and send a STUN/UDP/IP packet in reply with the ECN field set to not-ECT and an ECN-CHECK attribute included. A STUN server that doesn't understand the extension, or is incapable of reading the ECN values on incoming STUN packets, should follow the rule in the STUN specification for unknown comprehension-optional attributes and ignore the attribute, resulting in the sender receiving a STUN response without the ECN-CHECK STUN attribute.

The ECN STUN checks can be lost on the path, for example, due to the ECT marking but also due to various other non ECN-related reasons causing packet loss. The goal is to detect when the ECT markings are rewritten or if it is the ECT marking that causes packet loss so that the path can be determined as not-ECT. Other reasons for packet loss should not result in a failure to verify the path as ECT. Therefore, a number of retransmissions should be attempted. But, the sender of ECN STUN checks will also have to set a criteria for when it gives up testing for ECN capability on the path. Since the ICE agent has successfully verified the path, an RTT measurement for this path can be performed. To have a high probability of successfully verifying the path, it is RECOMMENDED that the client retransmit the ECN STUN check at least 4 times. The transmission for that flow is stopped when an ECN-CHECK STUN response has been received, which doesn't indicate a retransmission of the request due to a temporary error, or the maximum number of retransmissions has been sent. The ICE agent is recommended to give up on the ECN verification MAX(1.5*RTT, 20 ms) after the last ECN STUN check was sent.

The transmission of the ECT-marked STUN connectivity checks containing the ECN-CHECK attribute can be done prior as well in parallel to actual media transmission. Both cases are supported, where the main difference is how aggressively the transmission of the STUN checks are done. The reason for this is to avoid adding additional startup delay until media can flow. If media is required immediately after nomination has occurred, the STUN checks SHALL be done in parallel. If the application does not require media transmission immediately, the verification of ECT SHOULD start using the aggressive mode. At any point in the process until ECT has been verified or found to not work, media transmission MAY be started, and the ICE agent SHALL transition from the aggressive mode to the parallel mode.

The aggressive mode uses an interval between the retransmissions based on the Ta timer as defined in Section 16.1 for RTP Media Streams in ICE [RFC5245]. The number of ECN STUN checks needing to be sent will depend on the number of ECN-capable flows (N) that is to be established. The interval between each transmission of an ECN-CHECK packet MUST be Ta. In other words, for a given flow being verified for ECT, the retransmission timeout (RTO) is set to Ta*N.

The parallel mode uses transmission intervals in order to prevent the ECT verification checks from increasing the total bitrate more than 10%. As ICE’s regular transmission schedule is mimicking a common voice call in amount, to meet that goal for most media flows, setting the retransmission interval to Ta*N*k where k=10 fulfills that goal. Thus, the default behaviour SHALL be to use k=10 when in parallel mode. In cases where the bitrate of the STUN connectivity checks can be determined, they MAY be sent with smaller values of k, but k MUST NOT be smaller than 1, as long as the total bitrate for the connectivity checks are less than 10% of the used media bitrate. The RTP media packets being sent in parallel mode SHALL NOT be ECT marked prior to verification of the path as ECT.

The STUN ECN-CHECK attribute contains one field and a flag, as shown in Figure 6. The flag indicates whether the echo field contains a valid value or not. The field is the ECN echo field and, when valid, contains the two ECN bits from the packet it echoes back. The ECN-CHECK attribute is a comprehension optional attribute.

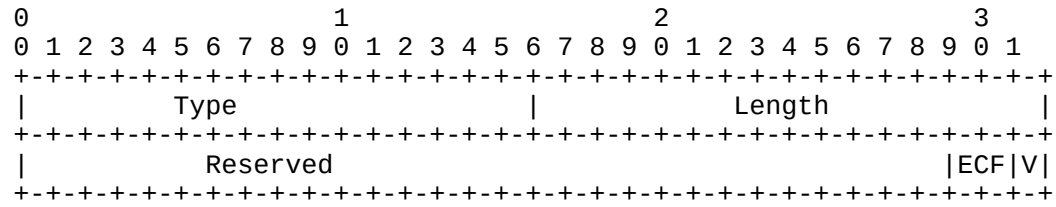

Figure 6: ECN-CHECK STUN Attribute

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

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

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

This attribute MAY be included in any STUN request to request the ECN field to be echoed back. In STUN requests, the V bit SHALL be set to 0. A compliant STUN server receiving a request with the ECN-CHECK attribute SHALL read the ECN field value of the IP/UDP packet in which the request was received. Upon forming the response, the server SHALL include the ECN-CHECK attribute setting the V bit to valid and include the read value of the ECN field into the ECF field. If the STUN responder was unable to ascertain, due to temporary errors, the ECN value of the STUN request, it SHALL set the V bit in the response to 0. The STUN client may retry immediately.

The ICE-based initialisation method does require some special consideration when used by a translator. This is especially for transport translators and translators that fragment or reassemble packets, since they do not separate the ECN control loops between the endpoints and the translator. When using ICE-based initiation, such a translator must ensure that any participants joining an RTP session for which ECN has been negotiated are successfully verified in the direction from the translator to the joining participant. Alternatively, it must correctly handle remarking of ECT RTP packets towards that participant. When a new participant joins the session, the translator will perform a check towards the new participant. If that is successfully completed, the ECT properties of the session are maintained for the other senders in the session. If the check fails, then the existing senders will now see a participant that fails to receive ECT. Thus, the failure detection in those senders will eventually detect this. However, to avoid misusing the network on the path from the translator to the new participant, the translator

SHALL remark the traffic intended to be forwarded from ECT to not-ECT. Any packets intended to be forwarded that are ECN-CE marked SHALL be discarded and not sent. In cases where the path from a new participant to the translator fails the ECT check, then only that sender will not contribute any ECT-marked traffic towards the translator.

7.2.3. Leap-of-Faith ECT Initiation Method

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

If the sender marks all packets as ECT while transmitting on a path that contains an ECN-blocking middlebox, then receivers downstream of that middlebox will not receive any RTP data packets from the sender and hence will not consider it to be an active RTP SSRC. The sender can detect this and revert to sending packets without ECT marks, since RTCP SR/RR packets from such receivers will either not include a report for the sender’s SSRC or will report that no packets have been received, but this takes at least one RTCP reporting interval. It should be noted that a receiver might generate its first RTCP packet immediately on joining a unicast session, or very shortly after joining an RTP/AVPF session, before it has had chance to receive any data packets. A sender that receives an RTCP SR/RR packet indicating lack of reception by a receiver SHOULD therefore wait for a second RTCP report from that receiver to be sure that the lack of reception is due to ECT-marking. Since this recovery process can take several tens of seconds, during which time the RTP session is unusable for media, it is NOT RECOMMENDED that the leap-of-faith ECT initiation method be used in environments where ECN-blocking middleboxes are likely to be present.

7.3. Ongoing Use of ECN within an RTP Session

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

7.3.1. Transmission of ECT-Marked RTP Packets

After a sender has successfully initiated ECN use, it SHOULD mark all the RTP data packets it sends as ECT. The sender SHOULD mark packets as ECT(0) unless the receiver expresses a preference for ECT(1) or for a random ECT value using the "ect" parameter in the "a=ecn-capable-rtp:" attribute.

The sender SHALL NOT include ECT marks on outgoing RTCP packets and SHOULD NOT include ECT marks on any other outgoing control messages (e.g., STUN [RFC5389] packets, Datagram Transport Layer Security (DTLS) [RFC6347] handshake packets, or ZRTP [RFC6189] control packets) that are multiplexed on the same UDP port. For control packets there might be exceptions, like the STUN-based ECN-CHECK defined in Section 7.2.2.

7.3.2. Reporting ECN Feedback via RTCP

An RTP receiver that receives a packet with an ECN-CE mark, or that detects a packet loss, MUST schedule the transmission of an RTCP ECN feedback packet as soon as possible (subject to the constraints of [RFC4585] and [RFC3550]) to report this back to the sender unless no timely feedback is required. The feedback RTCP packet SHALL consist of at least one ECN feedback packet (Section 5.1) reporting on the packets received since the last ECN feedback packet and will contain (at least) an RTCP SR/RR packet and an SDES packet, unless reduced-size RTCP [RFC5506] is used. 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 endpoints. Both RTP/AVPF and reduced-size RTCP MUST be negotiated in the session setup signalling before they can be used.

Every time a regular compound RTCP packet is to be transmitted, an ECN-capable RTP receiver MUST include an RTCP XR ECN Summary Report as described in Section 5.2 as part of the compound packet.

The multicast feedback implosion problem, which occurs when many receivers simultaneously send feedback to a single sender, must 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.

7.3.3. Response to Congestion Notifications

The reception of RTP packets with ECN-CE marks in the IP header is a notification that congestion is being experienced. The default reaction on the reception of these ECN-CE-marked packets MUST be to provide the congestion control algorithm with a congestion notification that triggers the algorithm to react as if packet loss had occurred. There should be no difference in congestion response if ECN-CE marks or packet drops are detected.

Other reactions to ECN-CE may be specified in the future, following IETF Review. Detailed designs of such alternative reactions MUST be specified in a Standards Track RFC and be reviewed to ensure they are safe for deployment under any restrictions specified. A potential example for an alternative reaction could be emergency communications (such as that generated by first responders, as opposed to the general public) in networks where the user has been authorised. A more detailed description of these other reactions, as well as the types of congestion control algorithms used by end-nodes, is outside the scope of this document.

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 is responsible for adapting the transmitted bitrate in response to RTCP ECN feedback. When the sender receives the ECN feedback data, it feeds this information into its congestion control or bitrate adaptation mechanism so that it can react as if packet loss 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: In a receiver-driven congestion control mechanism, the receivers can react to the ECN-CE marks themselves without providing ECN-CE feedback to the sender. This may allow faster response than sender-driven congestion control in some circumstances and also scale to large number of receivers and multicast usage. One example of receiver-driven congestion control is 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 packets to check if it experiences congestion with the current number of layers. If congestion is experienced, the receiver drops one layer, thus reducing the resource consumption on the path towards itself. For example, if a layered media encoding scheme such as H.264 Scalable Video Coding (SVC) is used, the receiver may change its layer

subscription and so reduce the bitrate it receives. The receiver MUST still send an RTCP XR ECN Summary to the sender, even if it can adapt without contact with the sender, so that the sender can determine if ECN is supported on the network path. The timeliness of RTCP feedback is less of a concern with receiver-driven congestion control, and regular RTCP reporting of ECN summary information is sufficient (without using RTP/AVPF immediate or early feedback).

Hybrid: There might be mechanisms that utilise both some receiver behaviours and some sender-side monitoring, thus requiring both feedback of congestion events to the sender and taking receiver decisions and possible signalling to the sender. In this case, the congestion control algorithm needs to use the signalling to indicate which features of ECN for RTP are required.

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 receivers having to settle for the lowest common rate.

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

7.4. Detecting Failures

Senders and receivers can deliberately ignore ECN-CE and thus get a benefit over behaving flows (cheating). The ECN nonce [RFC3540] is an addition to TCP that attempts to solve this issue as long as the sender acts on behalf of the network. The assumption that senders

act on behalf of the network may be false due to the nature of peer-to-peer use of RTP. Still, a significant portion of RTP senders are infrastructure devices (for example, streaming media servers) that do have an interest in protecting both service quality and the network. Even though there may be cases where the nonce may be applicable for RTP, it is not included in this specification. This is because a receiver interested in cheating would simply claim to not support the nonce, or even ECN itself. It is, however, worth mentioning that, as real-time media is commonly sensitive to increased delay and packet loss, it will be in the interest of both the media sender and receivers to minimise the number and duration of any congestion events as they will adversely 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 an effect on the ECN and application functionality. The gravest is if the node drops packets with the ECN field set to ECT(0), ECT(1), or ECN-CE. This can be detected by the receiver when it receives an RTCP SR packet indicating that a sender has sent a number of packets that it has not received. The sender may also detect such a middlebox based on the receiver's RTCP RR packet, when the extended sequence number is not advanced due to the failure to receive packets. If the packet loss is less than 100%, then packet loss reporting in either the ECN feedback information or RTCP RR will indicate the situation. The other action is to re-mark a packet from ECT or ECN-CE to not-ECT. That has less dire results; however, it should be detected so that ECN usage can be suspended to prevent misusing the network.

The RTCP XR ECN summary packet and the ECN feedback packet allow the sender to compare the number of ECT-marked packets of different types received with the number it actually sent. The number of ECT packets received, plus the number of ECN-CE-marked and lost packets, should correspond to the number of sent ECT-marked packets plus the number of received duplicates. If these numbers don't agree, there are two likely reasons: a translator changing the stream or not carrying the ECN markings forward or some node re-marking 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 behaviour is happening on the path.

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

7.4.1. Fallback Mechanisms

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

A receiver that detects a packet loss burst MAY schedule an early feedback packet that includes at least the RTCP RR and the ECN feedback message to report this to the sender. This will speed up the detection of the loss at the sender, thus triggering sender-side mitigation.

A sender that detects high packet loss rates for ECT-marked packets SHOULD immediately switch to sending packets as not-ECT to determine if the losses are potentially 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 two possible actions exist:

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

2. Renegotiate the session to disable ECN support. This is a choice that is suitable if the impact of ECT probing on the media quality is noticeable. If multiple initiations have 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 select to deliver media from the sender only intermittently; load-balancing devices that may in worst case result in some packets taking a different network path than the others; mobility solutions that switch the underlying network path in a transparent way for the sender or receiver; and membership changes in a multicast group. It is, however, appropriate to mention that there are also issues such as re-routing of traffic due to a flappy route table or excessive reordering and other issues that are not directly ECN related but nevertheless may cause problems for ECN.

7.4.2. Interpretation of ECN Summary Information

This section contains discussion on how the ECN Summary Report information can be used to detect various types of ECN path issues. We first review the information the RTCP reports provide on a per-source (SSRC) basis:

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

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.

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

Lost Packets Counter: The number of RTP packets that where expected based on sequence numbers but never received.

Duplication Counter: The number of received RTP packets that are duplicates of already received ones.

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 initialised to zero to provide values for the RTP stream sender from the first report. After the first report, the changes between the last received report and the previous report are determined by simply taking the values of the latest minus the previous, taking 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 ECN-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 rate of packet duplication is tracked, allowing one to take the duplication into account. The value of the ECN field for duplicates will also be counted, and when comparing the figures, one needs to take into account in the calculation that some fraction of packet duplicates are not-ECT and some are ECT. Thus, when only sending not-ECT, the number of sent packets plus reported duplicates equals the number of received not-ECT. When sending only ECT, the number of sent ECT packets plus duplicates will equal ECT(0), ECT(1), ECN-CE, and packet loss. When sending a mix of not-ECT and ECT, there is an uncertainty if any duplicate or packet loss was an not-ECT or ECT. If the packet duplication is completely independent of the usage of

ECN, then the fraction of packet duplicates should be in relation to the number of not-ECT vs. ECT packets sent during the period of comparison. This relation does not hold for packet loss, where higher rates of packet loss for not-ECT is expected than for ECT traffic.

Detecting clearing of ECN field: If the ratio between ECT and not-ECT transmitted in the reports has become all not-ECT, or has substantially changed towards not-ECT, then this is clearly an 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 is low in either of the categories. One must also take into account the level of CE marking. A CE-marked packet would have been dropped unless it was ECT marked. Thus, the packet loss level for not-ECT should be approximately equal to the loss rate for ECT when counting the CE-marked packets as lost ones. A sender performing this calculation needs to ensure that the difference is statistically significant.

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

8. Processing ECN in RTP Translators and Mixers

RTP translators and mixers that support ECN for RTP are required to process and potentially modify or generate ECN marking in RTP packets. They also need to process and potentially modify or generate RTCP ECN feedback packets for the translated and/or mixed streams. This includes both downstream RTCP reports generated by the media sender and also reports generated by the receivers, flowing upstream back towards the sender.

8.1. Transport Translators

Some translators only perform transport-level translations, such as copying packets from one address domain, like from unicast to multicast. They may also perform relaying like copying an incoming packet to a number of unicast receivers. This section details the ECN-related actions for RTP and RTCP.

For RTP data packets, the translator, which does not modify the media stream, SHOULD copy the ECN bits unchanged from the incoming to the outgoing datagrams, unless the translator itself is overloaded and experiencing congestion, in which case it may mark the outgoing datagrams with an ECN-CE mark.

A transport translator does not modify RTCP packets. However, it MUST perform the corresponding transport translation of the RTCP packets as it does with RTP packets being sent from the same source/ endpoint.

8.2. Fragmentation and Reassembly in Translators

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

The processing of arriving RTP packets for this case is as follows. If an ECN-marked packet is split into two, then both the outgoing packets MUST be ECN marked identically to the original; if several ECN-marked packets are combined into one, the outgoing packet MUST be either ECN-CE marked or dropped if any of the incoming packets are ECN-CE marked. If the outgoing combined packet is not ECN-CE marked, then it MUST be ECT marked if any of the incoming packets were ECT marked.

RTCP ECN feedback packets (Section 5.1) contain seven fields that are rewritten in an RTP translator that fragments or reassembles packets: the extended highest sequence number, the duplication counter, the Lost Packets Counter, the ECN-CE counter, and not-ECT counter, the ECT(0) counter, and the ECT(1) counter. The RTCP XR report block for ECN summary information (Section 5.2) includes all of these fields except the extended highest sequence number, which is present in the

report block in an SR or RR packet. The procedures for rewriting these fields are the same for both the RTCP ECN feedback packet and the RTCP XR ECN summary packet.

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

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

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

The ratio, R, may have a value that leads to non-integer multiples of the counters when translating the RTCP packet. For example, a Voice over IP (VoIP) translator that combines two adjacent RTP packets into one if they contain active speech data, but passes comfort noise packets unchanged, would have an R value of between 0.5 and 1.0 depending on the amount of active speech. Since the counter values in the translated RTCP report are integer values, rounding will be necessary in this case.

When rounding counter values in the translated RTCP packet, the translator should try to ensure that they sum to the number of RTP packets in the pre-translation sequence number space (numOrig). The

translator should also try to ensure that no non-zero counter is rounded to a zero value, unless the pre-translated values are zero, since that will lose information that a particular type of event has occurred. It is recognised that it may be impossible to satisfy both of these constraints; in such cases, it is better to ensure that no non-zero counter is mapped to a zero value, since this preserves congestion adaptation and helps the RTCP-based ECN initiation process.

One should be aware of the impact this type of translator has on the measurement of packet duplication. A translator performing aggregation and most likely also an fragmenting translator will suppress any duplication happening prior to itself. Thus, the reports and what is being scaled will only represent packet duplication happening from the translator to the receiver reporting on the flow.

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

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

8.3. Generating RTCP ECN Feedback in Media Transcoders

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

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

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

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

8.4. Generating RTCP ECN Feedback in Mixers

An RTP mixer terminates one-or-more RTP flows, combines them into a single outgoing media stream, and transmits that new stream as a separate RTP flow. A mixer has its own SSRC and is visible to other participants in the session at the RTP layer.

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

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

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

9. Implementation Considerations

To allow the use of ECN with RTP over UDP, an RTP implementation desiring to support receiving ECN-controlled media streams must support reading the value of the ECT bits on received UDP datagrams, and an RTP implementation desiring to support sending ECN-controlled media streams must support setting the ECT bits in outgoing UDP datagrams. The standard Berkeley sockets API pre-dates the specification of ECN and does not provide the functionality that is required for this mechanism to be used with UDP flows, making this specification difficult to implement portably.

10. IANA Considerations

10.1. SDP Attribute Registration

Following the guidelines in [RFC4566], the IANA has registered one new media-level SDP attribute:

- Contact name, email address and telephone number: Authors of RFC 6679
- 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) for RTP flows running over UDP/IP. This attribute is put in the SDP offer if the offering party wishes to receive an ECT flow. The answering party then includes the attribute in the answer if it wishes to receive an ECT flow. If the answerer does not include the attribute, then ECT MUST be disabled in both directions.

10.2. RTP/AVPF Transport-Layer Feedback Message

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

Name:	RTCP-ECN-FB
Long name:	RTCP ECN Feedback
Value:	8
Reference:	RFC 6679

10.3. RTCP Feedback SDP Parameter

The IANA has registered one new SDP "rtcp-fb" attribute "nack" parameter "ecn" in the SDP ("ack" and "nack" Attribute Values) registry.

```
Value name:     ecn
Long name:      Explicit Congestion Notification
Usable with:    nack
Reference:      RFC 6679
```

10.4. RTCP XR Report Blocks

The IANA has registered one new RTCP XR Block Type as defined in Section 5.2:

```
Block Type: 13
Name:       ECN Summary Report
Reference:  RFC 6679
```

10.5. RTCP XR SDP Parameter

The IANA has registered one new RTCP XR SDP Parameter "ecn-sum" in the "RTCP XR SDP Parameters" registry.

Parameter name	XR block (block type and name)
ecn-sum	13 ECN Summary Report

10.6. STUN Attribute

A new STUN [RFC5389] attribute in the comprehension-optional range under IETF Review (0x8000-0xFFFF) has been assigned to the ECN-CHECK STUN attribute (0x802D) defined in Section 7.2.2. The STUN attribute registry can currently be found at: http://www.iana.org/assignments/stun-parameters.

10.7. ICE Option

A new ICE option "rtp+ecn" has been registered in the "ICE Options" registry created by [RFC6336].

11. Security Considerations

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

External threats to the RTP and RTCP traffic:

Denial of Service affecting RTCP: 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; or 2) ensure that the sender disables the usage of ECN by reporting failures to receive ECN by changing the counter fields. This can also be accomplished by injecting false RTCP packets to the media sender. Reporting a lot of ECN-CE-marked traffic is likely the more efficient denial-of-service tool as that may likely force the application to use the lowest possible bitrates. The prevention against an external threat is to integrity protect the RTCP feedback information and authenticate the sender.

Information leakage: The ECN feedback mechanism exposes the receiver's perceived packet loss and the packets it considers to be ECN-CE marked. This is mostly not considered sensitive information. If it is considered sensitive, the RTCP feedback should be encrypted.

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

Denial of Service affecting the session setup 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.

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

bitrate share across the resource bottleneck. It is far from certain that a receiver would be able to get a significant larger share of the resources. That assumes a high enough level of aggregation that there are flows to acquire shares from. The risk of cheating is that failure to react to congestion results in packet loss and increased path delay.

Receivers misbehaving: A receiver may prevent the usage of ECN in an RTP session by reporting itself as non-ECN capable, 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 sessions, some motivation exists for why a receiver would misbehave as it can prevent 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 it no longer has any valid security context to affect the session.

Misbehaving senders: The enabling of ECN gives the media packets a higher degree of probability to reach the receiver compared to not-ECT-marked ones on an ECN-capable path. However, this is no magic bullet, and failure to react to congestion will most likely only slightly delay a network buffer over-run, in which its session also will experience packet loss and increased delay. There is some possibility that the media sender’s traffic will push other traffic out of the way without being affected too 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 an unfair share. This can only be detected and potentially prevented by network monitoring and administrative entities. See Section 7 of [RFC3168] for more discussion of this issue.

We note that the endpoint security functions needed to prevent an external attacker from interfering with the signalling are source authentication and integrity protection. To prevent information leakage from the feedback packets, encryption of the RTCP is also needed. For RTP, multiple possible solutions exist depending on the application context. Secure RTP (SRTP) [RFC3711] does satisfy the requirement to protect this mechanism. Note, however, that when using SRTP in group communication scenarios, different parties might

share the same security context; in this case, the authentication mechanism only shows that one of those parties is involved, not necessarily which one. IPsec [RFC4301] and DTLS [RFC6347] can also provide the necessary security functions.

The signalling protocols used to initiate an RTP session also need to be source authenticated and integrity protected to prevent an external attacker from modifying any signalling. An appropriate mechanism to protect the used signalling needs to be used. For SIP/SDP, ideally Secure MIME (S/MIME) [RFC5751] would be used. However, with the limited deployment, a minimal mitigation strategy is to require use of SIPS (SIP over TLS) [RFC3261] [RFC5630] to at least accomplish hop-by-hop protection.

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

12. Examples of SDP Signalling

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

12.1. Basic SDP Offer/Answer

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

The Offer:

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

This SDP offer presents a single media stream with 3 media payload types. It proposes to use ECN with RTP, with the ICE-based initialisation being preferred over the RTP/RTCP one. Leap of faith is not suggested to be used. The offerer is capable of both setting and reading the ECN bits. In addition, the use of both the RTCP ECN feedback packet and the RTCP XR ECN Summary Report are supported. ICE is also proposed with two candidates. It also supports reduced-size RTCP and can use it.

The Answer:

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

The answer confirms that only one media stream will be used. One RTP payload type was removed. ECN capability was confirmed, and the initialisation method will be ICE. However, the answerer is only capable of reading the ECN bits, which means that ECN can only be used for RTP flowing from the offerer to the answerer. ECT always set to 0 will be used in both directions. Both the RTCP ECN feedback packet and the RTCP XR ECN Summary Report will be used. Reduced-size RTCP will not be used as the answerer has not indicated support for it in the answer.

12.2. Declarative Multicast SDP

The session below describes an Any-Source Multicast using a session with a single media stream.

```
v=0
o=jdoe 3502844782 3502844782 IN IP4 198.51.100.235
s=Multicast SDP session using ECN for RTP
i=Multicasted audio chat using ECN for RTP
b=AS:128
t=3502892703 3502910700
m=audio 56144 RTP/AVPF 97
c=IN IP4 233.252.0.212/127
a=rtpmap:97 g719/48000/1
a=fmtp:97 maxred=160
a=maxptime:160
a=ptime:20
a=ecn-capable-rtp: rtp mode=readonly; ect=0
a=rtcp-fb:* nack ecn
a=rtcp-fb:* trr-int 1500
a=rtcp-xr:ecn-sum
```

This is a declarative SDP example and indicates required functionality in the consumer of the SDP. The initialisation method required is the RTP/RTCP-based one, indicated by the "a=ecn-capable-rtp: rtp ..." line. Receivers are required to be able to read ECN marks ("mode=readonly"), and the ECT value is recommended to be set to 0 always ("ect=0"). The ECN usage in this session requires both ECN feedback and RTCP XR ECN Summary Reports, and their use is indicated through the "a=rtcp-fb:" and "a=rtcp-xr:ecn-sum" lines.

13. Acknowledgments

The authors wish to thank the following individuals for their reviews and comments: Thomas Belling, Bob Briscoe, Roni Even, Kevin P. Flemming, Tomas Frankkila, Christian Groves, Christer Holmgren, Cullen Jennings, Tom Van Caenegem, Simo Veikkolainen, Bill Ver Steeg, Dan Wing, Qin Wu, and Lei Zhu.

14. References

14.1. 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.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

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

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

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

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

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

[RFC6336] Westerlund, M. and C. Perkins, "IANA Registry for Interactive Connectivity Establishment (ICE) Options", RFC 6336, July 2011.

14.2. Informative References

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, August 1989.

[RFC2762] Rosenberg, J. and H. Schulzrinne, "Sampling of the Group Membership in RTP", RFC 2762, February 2000.

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

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

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

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

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

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

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

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

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

[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.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, July 2006.

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

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

[RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.

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

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

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

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

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

[RFC6189] Zimmermann, P., Johnston, A., and J. Callas, "ZRTP: Media Path Key Agreement for Unicast Secure RTP", RFC 6189, April 2011.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

Authors' Addresses

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden
Phone: +46 10 714 82 87
EMail: magnus.westerlund@ericsson.com

Ingemar Johansson
Ericsson
Laboratoriegrand 11
SE-971 28 Lulea
Sweden
Phone: +46 73 0783289
EMail: ingemar.s.johansson@ericsson.com

Colin Perkins
University of Glasgow
School of Computing Science
Glasgow G12 8QQ
United Kingdom
EMail: csp@csperkins.org

Piers O'Hanlon
University of Oxford
Oxford Internet Institute
1 St Giles
Oxford OX1 3JS
United Kingdom
EMail: piers.ohanlon@oii.ox.ac.uk

Ken Carlberg
G11
1600 Clarendon Blvd
Arlington, VA
USA
EMail: carlberg@g11.org.uk